UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Priscilla Kelly Machado Vieira

Recomendação Semântica de Conteúdo em Ambientes de Convergência Digital

JOÃO PESSOA
MARÇO DE 2013

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Priscilla Kelly Machado Vieira

Recomendação Semântica de Conteúdo em Ambientes de Convergência Digital

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba, como requisito parcial para a obtenção do título Mestre em Informática (Sistemas de Computação).

Linha de Pesquisa: Computação Distribuída

Orientadora: Prof. Drª. Natasha Correia Queiroz Lino.

JOÃO PESSOA
MARÇO DE 2013

V658r Vieira, Priscilla Kelly Machado.

Recomendação semântica de conteúdo em ambientes de convergência digital / Priscilla Kelly Machado Vieira.- João Pessoa, 2013.

128f. : il.

Orientadora: Natasha Correia Queiroz Lino

Dedico este trabalho aos meus pais, Magna e Ricardo, por todo amor, atenção e ensinamentos passados em minha vida.

Agradecimentos

A ajuda recebida foi imprescindível para a realização deste mestrado. Portanto, gostaria de externar sinceramente meus agradecimentos a todos que contribuíram para a conclusão desta etapa.

À Deus, por me dar forças para superar os obstáculos que surgem no caminho para se atingir objetivos.

Aos meus pais, José Ricardo e Magna Maria, pelo apoio e suporte em todos os momentos desta jornada.

À minha prima Maiane Machado e seu esposo Henrique César, por toda ajuda e acolhimento durante as atividades deste trabalho de mestrado.

Ao meu grande amigo e namorado, Wilker Victor, por todo incentivo, compreensão e paciência demonstrados ao longo deste período de dedicação aos estudos.

Aos colegas do projeto *Knowledge TV*, por todo companheirismo, apoio e dedicação, os quais foram fundamentais para o desenvolvimento do trabalho.

À professora Natasha Queiroz, pelas discussões, comentários e orientação valiosa, além da dedicação e incentivo ao aperfeiçoamento profissional.

À Capes, pela concessão da bolsa que permitiu o andamento tranquilo da pesquisa e o subsídio para divulgação do trabalho em congressos.

A todos que, direta ou indiretamente, contribuíram para a concretização deste trabalho.

Resumo

Com o advento da TV Digital interativa (TVDi), nota-se o aumento de interatividade no processo de comunicação além do incremento das produções audiovisuais, elevando o número de canais e recursos disponíveis para o usuário. Esta realidade faz da tarefa de encontrar o conteúdo desejado uma ação onerosa e possivelmente ineficaz. A incorporação de sistemas de recomendação no ambiente TVDi emerge como uma possível solução para este problema. Este trabalho tem como objetivo propor uma abordagem híbrida para recomendação de conteúdo em TVDi, baseada em técnicas de Mineração de Dados, integradas a conceitos da Web Semântica, permitindo a estruturação e padronização dos dados e consequente possibilidade do compartilhamento de informações, provendo semântica e raciocínio automático. Para o serviço proposto é considerado o Sistema Brasileiro de TV Digital e o *middleware* Ginga. Foi desenvolvido um protótipo e realizado experimentos com a base de dados do NetFlix, utilizando a métrica de precisão para avaliação. Obteve-se uma precisão média de 30%, utilizando apenas a técnica de mineração. Acoplando-se com as regras semânticas obteve-se precisão média de 35%.

Palavras-chave: TV Digital interativa; Sistemas de Recomendação; Ginga; Web Semântica, Mineração de Dados;

Abstract

The emerging scenario of interactive Digital TV (iDTV) is promoting the increase of interactivity in the communication process and also in audiovisual production, thus rising the number of channels and resources available to the user. This reality makes the task of finding the desired content becoming a costly and possibly ineffective action. The incorporation of recommender systems in the iDTV environment is emerging as a possible solution to this problem. This work aims to propose a hybrid approach to content recommendation in iDTV, based on data mining techniques, integrated the concepts of the Semantic Web, allowing structuring and standardization of data and consequent possibility of sharing information, providing semantics and automated reasoning. For the proposed service is considered the Brazilian Digital TV System and the middleware Ginga. A prototype has been developed and carried out experiments with NetFlix database using the measuring accuracy for evaluation. There was obtained an average accuracy of 30% using only mining technique. Including semantic rules obtained average accuracy of 35%.

Keywords: *Interactive Digital TV; Recommender Systems TV Digital Interativa*; *Ginga, Semantic Web, Data Mining;*

Lista de Ilustrações

Lista de Tabelas

Lista de Siglas

Sigla	Significado
ATSC	*Advanced Television Systems Committee*
CTIC	Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para a Informação e Comunicação
DMOP	*Data Mining Optimization*
DTMB	*Digital Terrestrial Multimedia Broadcast*
DVB	*Digital Video Broadcast*
EPG	*Electronic Programming Guide*
ETL	*Extract Transformation and Load*
ISDB	*Integrated Services Digital Broadcasting*
ISDTV-T	*International Standard for Digital Television Terrestrial*
ITU	*International Telecomunication Union*
KDD	*Knowledge Discovery in Databases*
KTV	*Knowledge TV*
PSI	*Program Specific Information*
RC	Representação de Conhecimento
RKTV	*Recommender Knowledge TV*
SBTVD	Sistema Brasileiro de TV Digital
SI	*Service Information*
STB	*Set Top Box*
NCL	*Nested Context Language*
OntoClusteringKTV	Ontologia *Clustering Knowledge TV*
OntoRKTV	Ontologia *Recommender Knowledge TV*

OntoMKTV	Ontologia *Mining Knowledge TV*
OWL	*Web Ontology Language*
RDF	*Resource Description Language*
TF-IDF	*Term Frequency – Inverse Document Frequency*
TVA	*TV-Anytime*
TVDi	TV Digital interativa
UFPB	Universidade Federal da Paraíba
XML	*eXtensible Markup Language*
W3C	*World Wide Web Consortium*

Sumário

Capítulo 1

Introdução

A evolução tecnológica associada com a inserção da TV Digital (TVD) no Brasil propicia transformações profundas em todas as instâncias relativas ao meio, caracterizado por um processo de convergência a uma nova plataforma de comunicação baseada em tecnologias digitais de codificação, compressão, multiplexação, transporte, transmissão e recepção de informações. Adicionalmente, a TV Digital interativa (TVDi) (Lugmayr et *al*., 2004) congrega uma gama de possibilidades que incrementam o processo de comunicação: (i) Inclui a interatividade e (ii) Possibilita conectividade com a Web (TVDi conectada). Estas características redemocratizam o acesso à informação (Médola, 2009), principalmente se considerarmos que a TV ainda é o principal meio de transmissão de informações e entretenimento para a população brasileira, estando presente em mais de 95% das residências; enquanto que os computadores possuem uma cobertura de pouco mais de 38% das residências brasileiras (IBGE, 2012).

Com o advento da TV Digital interativa (TVDi), nota-se o aumento de interatividade no processo de comunicação além do incremento das produções audiovisuais (Médola, 2009), o que fomenta um ambiente favorável ao desenvolvimento de aplicações dedicadas a esta tecnologia. Neste sentido, a TVDi promoveu um cenário de aumento da quantidade de canais, serviços e conteúdo disponíveis ao usuário. Esta nova realidade faz com que a tarefa de encontrar o conteúdo desejado se torne uma ação onerosa e, em alguns casos, ineficiente. É neste contexto que Sistemas de Recomendação (Resnick & Varian, 1997) emergem como solução possível para auxiliar esta escolha.

1.1 Motivação

Sistemas de recomendação são comumente aplicados no *e-commerce,* a exemplo da recomendação de produtos na *Amazon.com* (Linden et *al.*, 2003). Mais recentemente começou-se o estudo de recomendação de conteúdo no ambiente da TVDi, no entanto, algumas peculiaridades diferem a TVDi em relação ao ambiente Web e estas características devem ser avaliadas ao se desenvolver um sistema de recomendação para TVDi.

A TVDi é caracterizada por ser um ambiente (i) multiusuário, onde grupos de pessoas se reúnem para consumir seu conteúdo (Chorianopoulos, 2008), e (ii) monousuário, onde a experiência é vivenciada por uma única pessoa. Desta forma, as abordagens de recomendação de conteúdo devem ser adequadas ao tipo de experiência dos usuários, sendo mais desafiadora a experiência multiusuário, focando no conteúdo visto pelo grupo e identificando particularidades de uso.

Outro desafio para sistemas de recomendação para TVDi é a coleta de dados (i) Implícita e/ou (ii) Explícita (Chorianopoulos, 2008). A coleta explícita necessita que o usuário informe claramente suas preferências por meio de uma interação direta, o que torna o processo muitas vezes dispendioso, sendo dificultado com a forma de entrada de dados via controle remoto. Esta abordagem pode não ser a mais adequada para o domínio da TV, por ser caracterizado, predominantemente, por interações passivas dos usuários (Chorianopoulos, 2008). Em contrapartida, a coleta de dados implícita é transparente ao usuário, onde seu comportamento é monitorado por meio do histórico do sistema, sendo mais adequada, portanto, ao ambiente da TV.

Com a TVDi espera-se uma mudança no comportamento do usuário quanto ao modo de se portar em relação aos conteúdos visualizados na TV (Montez & Becker, 2005). O telespectador tem a possibilidade de interagir com o transmissor de modo ativo, permitindo a bidirecionalidade de informações e o desenvolvimento de diversas aplicações como: jogos, *T-commerce*, *T-learning*, *T-government*, tal como o desenvolvimento de sistemas de recomendação de conteúdo.

Essas características foram observadas para a concepção da estratégia de recomendação utilizada neste trabalho, o qual propõe um módulo de recomendação adequado à arquitetura da TVDi brasileira, considerando-a como um ambiente de

convergência digital (TV e Web) (Lino et *al.*, 2011), inserido no *middleware* Ginga (Soares & Lemos, 2007), permitindo o direcionamento de conteúdo de acordo com o *Set-Top-Box* (STB).

O serviço de recomendação utiliza uma abordagem baseada em técnicas de Mineração de Dados (Han & Kamber, 2006), utilizando a coleta de dados implícita, integradas a conceitos da Web Semântica (Berners-Lee et *al.*, 2001), permitindo a estruturação e padronização dos dados e consequente possibilidade do compartilhamento de informações, provendo semântica e raciocínio automático.

No contexto da TV, recomendações podem almejar três diferentes públicos: (i) *Marketing*, para direcionar o conteúdo para usuários específicos, com objetivo de ampliar ou consolidar o mercado, (ii) Emissoras, para possibilitar a criação de grades de programação que maximizem as audiências e (iii) Usuários, para facilitar a escolha de conteúdo de acordo dom os hábitos de uso da TV. Neste trabalho o foco é a recomendação direcionada para usuários.

Na próxima Seção, serão definidos os Objetivos Gerais e Específicos para a consecução deste trabalho.

1.2 Objetivos

Diante da grande quantidade de conteúdo disponível na TVD e as dificuldades na seleção do conteúdo de interesse do usuário, sugere-se o desenvolvimento de um sistema de recomendação que auxilie nesta escolha. As técnicas utilizadas devem ser adequadas ao ambiente da TVD, proporcionando um melhor serviço. Neste sentido, seguem os objetivos do trabalho.

1.2.1 Objetivos Gerais

No contexto de sistemas de recomendação, diversas estratégias estão sendo utilizadas (Adomavicius & Tuzhilin, 2005). Este trabalho propõe o uso de técnicas de Mineração de Dados (Han & Kamber, 2006) para a detecção dos padrões de uso do *Set-Top-Box* (STB), resultado do uso coletivo do aparelho. Para isto, o processo de mineração de dados será realizado sobre o seu histórico de uso, utilizando coleta de dados implícita.

Observando as informações que podem ser extraídas após a mineração, propõe-se estruturar os dados minerados utilizando os conceitos e técnicas de

Representação de Conhecimento (RC) (Russell & Norvig, 2002), como por exemplo, Ontologias (Guarino, 1995), como também a integração de tecnologias da Web Semântica (Breitman, 2005) durante o processo de recomendação de conteúdo para a TVDi conectada. Esta abordagem permitirá a interoperabilidade de informações adquiridas, tal como o raciocínio automático para a geração de conhecimento.

Neste sentido, este trabalho tem como objetivo geral propor um sistema de recomendação adequado a arquitetura da TVDi conectada brasileira, interligando a Mineração de Dados aos conceitos da Web Semântica.

1.2.2 Objetivos Específicos

A fim de atingir o objetivo geral supracitado, os seguintes objetivos específicos se fizeram necessários:

- Realizar o levantamento do Estado da Arte em sistemas de recomendação para TV Digital;
- Permitir recomendação de conteúdo multimídia em TVDi conectada utilizando conceitos e técnicas da Web Semântica com Mineração de Dados;
- Possibilitar que o *middleware* Ginga forneça o serviço de recomendação à TVDi conectada;
- Obter um sistema de recomendação de conteúdo multimídia a partir do uso do STB;
- Propor uma arquitetura genérica de recomendação de conteúdo multimídia, possibilitando o serviço independente do ambiente provedor dos dados;
- Especificar uma ontologia para o processo de recomendação de conteúdo multimídia;
- Planejar Estudos de Caso;
- Avaliar a eficácia da proposta no ambiente da TVD.

1.3 Organização da Dissertação

O conteúdo deste trabalho está distribuído em 6 (seis) Capítulos. O Capítulo 2 (dois) aborda a Fundamentação Teórica tratando os conceitos básicos das principais áreas

temáticas que envolvem este trabalho: Sistemas de Recomendação de Conteúdo, Mineração de Dados, Web Semântica e TV Digital interativa.

No Capítulo 3 (três) é detalhado o processo de recomendação de conteúdo na plataforma KTV, enfatizando a arquitetura geral da plataforma e a interseção deste trabalho na plataforma.

O Capítulo 4 (quatro) apresenta detalhes da arquitetura proposta, tal como as ontologias construídas durante este trabalho de mestrado.

No Capítulo 5 (cinco) são apresentados os experimentos e cenários de uso para validação e discussão da arquitetura proposta.

O Capítulo 6 (seis) destina-se ao levantamento do estado da arte em Sistemas de Recomendação para TV Digital, apresentando um quadro comparativo entre as abordagens encontradas e este trabalho.

No Capítulo 7 (sete) são realizadas as Considerações Finais, discutidas as dificuldades encontradas durante a pesquisa e exposto os trabalhos futuros a este trabalho de mestrado.

Por fim são apresentadas as Referências Bibliográficas utilizadas neste trabalho.

Capítulo 2

Fundamentação Teórica

Nesta Seção serão apresentados os principais conceitos teóricos necessários para a compreensão deste trabalho. Inicialmente, serão explanados os principais conceitos de sistemas de recomendação. Em seguida, serão apresentados os conceitos da Web semântica. Por fim, serão explanados os conceitos relacionados a TVDi, focando na sua arquitetura e no *middleware* Ginga.

2.1 Sistemas de Recomendação

Sistemas de recomendação são responsáveis por conduzir o usuário, disponibilizando sugestões de acordo com o perfil deste usuário (Chorianopoulos, 2008). Assim sendo, produzem sugestões personalizadas, e guiam o usuário de forma individual nas escolhas de conteúdos de seu interesse (Burke, 2002).

Técnicas de recomendação são comumente aplicadas em *e-commerce,* a exemplo da loja da “Amazon” e do “Submarino”. Na Figura 1, é exemplificada a tela de recomendação de produtos da loja “Submarino”. É importante observar que a parte central da recomendação é personalizada para um usuário específico. No entanto, a parte inferior recomenda os produtos mais vendidos, independente de usuário. Destacando assim, duas formas diferentes de recomendar os produtos (Figura 1).

Figura 1 - Recomendação do site Submarino

Com o advento da TV Digital, e consequentemente, o aumento da interatividade no processo de comunicação, a redemocratização do acesso à informação (Médola, 2009), o maior incremento à produção audiovisual em níveis nacional e regional, a inclusão digital e a melhoria da qualidade da educação, alcançou-se a ampliação das possibilidades de serviços e canais de programações a serem oferecidos. Desta forma, a função de escolha de conteúdo tornou-se onerosa para o usuário. É neste contexto, considerando as características intrínsecas ao ambiente da TV, que sistemas de recomendação emergem como uma solução possível para auxiliar esta escolha.

Diversos aspectos devem ser considerados para a construção de um sistema de recomendação a exemplo de como as informações são coletadas e os métodos de recomendação empregados (Chorianopoulos, 2008). Estes aspectos são apresentados e discutidos nas subseções seguintes.

2.1.1 Coleta de Dados

Três abordagens são adotadas atualmente para coleta de informação: métodos explícitos, implícitos e híbridos.

- Coleta Explícita: Nesta abordagem o usuário informa claramente suas preferências (Xu et *al*., 2002). Pode ser realizada por meio de avaliações diretas do usuário e preenchimento de formulários com suas informações pessoais. A principal vantagem desta abordagem é a precisão para expressar preferências ou interesses. Por outro lado, a necessidade de interação direta torna o processo muitas vezes oneroso, sendo dificultado com a forma de entrada de dados via controle remoto. Para o domínio da TV, que é

caracterizado, predominantemente, por interações passivas, esta abordagem pode não ser a mais adequada (Chorianopoulos, 2008).

- Coleta Implícita: A coleta é transparente ao usuário; seu comportamento é monitorado por meio do histórico do sistema (Fernández et *al*., 2006). A principal vantagem desta abordagem é ser discreta ao objetivo do usuário, não sendo necessária sua interação direta com o sistema. Em contra partida, a precisão do sistema é atenuada, pois depende intrinsecamente dos dados coletados ao longo do uso do sistema.
- Coleta Híbrida: Esta abordagem é a combinação das duas anteriores. Explora avaliações explícitas sobre os programas a serem recomendados, por exemplo, e, de forma transparente, realiza as recomendações.

Para o domínio da TV, que é caracterizado, predominantemente, por interações passivas, a abordagem de coleta explícita pode não ser a mais adequada (Chorianopoulos, 2008). Sendo assim, esta proposta utilizará a abordagem Implícita para a coleta de dados.

2.1.2 Apresentação de Recomendação

Uma importante decisão a ser tomada em sistemas de recomendação é a forma que o conteúdo recomendado deve ser emitido para o usuário. Dentre as abordagens existentes, destacam-se: a *push* e a *pull*.

- *Push*: A apresentação das recomendações ocorre sem que o usuário solicite o serviço (Chorianopoulos, 2008). O usuário não necessita interagir com o sistema, no entanto pode estar recebendo um serviço indesejado.
- *Pull*: A apresentação das recomendações ocorre no momento em que o usuário solicita o serviço (Chorianopoulos, 2008). Esta abordagem proporciona maior controle do usuário sobre a aplicação, o tornando, assim, mais livre para decidir o momento que as recomendações o convêm.

2.1.3 Estratégias de Recomendação

Existem diversas formas de apresentar a recomendação ao indivíduo. As estratégias mais utilizadas definidas por Reategui & Cazella (2005) são: Listas de

Recomendação, Avaliações de Usuários, Recomendações Direcionadas, Associação de Itens e Associações por Conteúdo.

- Listas de Recomendações: Itens são organizados em listas, criando, assim, grupos de itens: (1) Itens mais vendidos, (2) Itens lançamentos, (3) Itens em promoção, dentre outras. Esta estratégia é de fácil implementação, largamente utilizada em comércio eletrônico e não necessita de informações do indivíduo. No entanto, impossibilita recomendações personalizadas.
- Avaliação de Usuários: Nesta estratégia, o sistema permite comentários dos indivíduos sobre os itens. Estes comentários podem aumentar a confiabilidade de novos indivíduos sobre o produto. Entre suas vantagens está a fácil implementação, no entanto há uma dependência da recomendação em relação à veracidade das opiniões dos indivíduos.
- Recomendações Direcionadas: Oferece uma seção de sugestões feitas para o usuário, a partir das informações coletadas de forma implícita (histórico do indivíduo) ou explícita (perfil do indivíduo). Esta estratégia pode ter algumas dificuldades de implementação, no entanto é mais personalizada que as citadas anteriormente.
- Associação de Itens: Utiliza técnicas para encontrar associação entre itens avaliados por usuários em uma base de dados. Trata-se de um método complexo, uma vez que é necessária uma análise profunda do comportamento do usuário, para que seja possível identificar padrões de uso.
- Associações por Conteúdo: O conteúdo dos itens é avaliado na geração das recomendações. Por exemplo, ao recomendar um livro, podem ser levados em conta atributos como autor, editora, gênero, etc.

Diferentes técnicas podem ser utilizadas para recomendação de conteúdo, algumas delas estão destacadas na Seção seguinte.

2.1.4 Categorização de Sistemas de Recomendação

Diversas técnicas tem surgido visando identificar padrões no comportamento dos usuários/clientes de sistemas computacionais. De forma geral, as técnicas e as

abordagens utilizadas determinam a classificação do sistema de recomendação. Tais classificações serão explanadas a seguir.

2.1.4.1 Categorização da Abordagem

Os sistemas de recomendação podem ser classificados de acordo com a abordagem de recomendação adotada. De acordo com Adomavicius & Tuzhilin (2005) as abordagens são classificadas em 3 (três) categorias: (i) Filtragem Baseada em Conteúdo (FBC), (ii) Filtragem Colaborativa (FC) e (iii) Filtragem Híbrida (FH).

I. Filtragem Baseada em Conteúdo

A FBC parte do princípio de que, se um usuário aprovou um determinado produto/programa, é provável que no futuro goste de outros similares, com as mesmas características (Herlocker, 2000). Técnicas enquadradas na FBC utilizam preferências anteriores do usuário, para inferir seu comportamento futuro.

Nesta abordagem de recomendação, é necessário calcular a similaridade entre os itens a serem recomendados. Desta forma, é necessário realizar a comparação entre os conteúdos dos itens.

Um algoritmo FBC bastante difundido é o TF-IDF (em inglês, *Term Frequency – Inverse Document Frequency*) (Salton, 1998). Em suma, o TF-IDF calcula a similaridade de textos, neste caso, descrições de itens, por meio da frequência de palavras chaves. Desta forma, o algoritmo é bastante eficiente em cenários nos quais os itens tem descrições textuais completas, sendo esta uma de suas limitações.

Um dos principais pontos de desvantagem da FBC é a dependência da descrição textual, sendo assim, itens diferentes com descrições semelhantes podem ser recomendados de forma equivocada. Além disto, podem ocorrer imprecisões em recomendações para novos usuários, dado que a FBC é baseada em usos passados dos itens.

II. Filtragem Colaborativa

A FC parte do princípio de que grupos de pessoas similares possuem comportamentos semelhantes (Torres, 2004), desta forma, pode existir uma troca de experiência entre usuários que possuem interesses comuns. Sendo assim, abordagens baseadas em FC tentam identificar grupos de indivíduos que possuem

hábitos semelhantes, estes indivíduos devem colaborar entre si com itens de seu uso, fazendo, assim, trocas de itens. Métodos que trabalham desta forma são chamados de procedimentos com vizinhança entre usuários e são, atualmente, os mais difundidos na comunidade (Sarwar, 2001).

As principais vantagens da FC são: (i) Não apresenta restrições quanto ao tipo de conteúdo que compõe os itens, como ocorre na FBC. (ii) Recomendações inesperadas são possíveis, pois itens nunca acessados por um individuo podem ser recomendados de forma precisa. Em contrapartida (Torres, 2004) define 3 (três) desvantagens para a FC: *First-rater problem, Startup problem e Sparcity problem*.

- *First-rater problem*: Primeiro avaliador, que acontece quando um novo item é adicionado na base de dados e ainda não há informações ao seu respeito, logo não será recomendado.
- *Startup problem*: Caracterizado pelo número pequeno de usuários, onde poucos usuários dificulta a descoberta de similaridade entre eles, diminuindo, assim, a precisão da recomendação.
- *Scarcity problem*: Caracterizado pela dispersão dos itens avaliados pelo o indivíduo em relação a todos os itens do sistema. Este problema ocorre em situações onde o número de itens é muito grande tornando impraticável as suas avaliações.

III. Filtragem Híbrida

A FH faz uso de mais de uma técnica de recomendação. Sistemas híbridos são interessantes, pois fazem combinações de técnicas que atenuam as desvantagens de cada um dos processos utilizados de forma isolada.

Para a construção de um Sistema Híbrido pode-se combinar de diversas formas um conjunto de técnicas. Devido a este grande número de possibilidades, algumas ainda não foram realizadas e avaliadas (Burke, 2002), sendo este fato um grande motivador para pesquisas neste âmbito.

Para a combinação de técnicas existe uma classificação definida em (Burke, 2002) destacada a seguir:

- *Weighted*: O resultado gerado por cada um das técnicas utilizadas tem um peso relacionado.
- *Mixed*: Intercala as recomendações do sistema híbrido por cada uma das técnicas de forma isolada.
- *Cascade*: Combina as técnicas que formam o sistema híbrido de forma que a recomendação gerada por uma técnica, de forma isolada, é refinada, posteriormente, por outra técnica e assim sucessivamente.
- *Switching*: Utiliza um critério para eleger uma das técnicas que compõem o sistema híbrido, como sendo a utilizada em um determinado momento de recomendação do sistema. Esta escolha é alterada para que a recomendação torne-se cada vez mais precisa.
- *Feature Combination*: Características de diferentes fontes de recomendação são implementadas em conjunto, em um único algoritmo de recomendação. Utiliza técnicas para adicionar dados em um item. Com isto, obtém-se uma melhora em recomendações com FBC.
- *Feature Argumentation*: Utiliza técnicas para enriquecer os dados já conhecidos de um item, complementando, desta forma, as informações ausentes.
- *Meta-level*: O modelo gerado por uma técnica, de forma isolada, é utilizado como entrada para a técnica posterior e assim sucessivamente.

As abordagens destacadas são elaboradas por meio de técnicas que manipulam informações para a recomendação. Dentre estas, está as de descoberta de conhecimento para apoio a recomendação.

2.1.4.2 Categorização da Técnica

Os sistemas de recomendação também podem ser classificados de acordo com a técnica de recomendação adotada. Neste trabalho, as dividimos em 3 (três) categorias: (i)Técnicas Baseadas nos Modelos Simbólicos, (ii)Técnicas Baseadas nos Modelos Conexionistas e (iii) Técnicas Probabilísticas.

I. Técnicas Baseadas nos Modelos Simbólicos

Sistemas baseados no modelo de raciocínio simbólico buscam aprender construindo representações simbólicas de um conceito por meio da análise de exemplos e contraexemplos desse conceito (Russell & Norvig, 2002). Dentre as técnicas baseadas em símbolos, neste trabalho destacamos 2 (duas): (a) Técnicas de Mineração de Dados e (b) Técnicas Baseadas em Semântica.

a. Técnicas Baseadas em Mineração de Dados

Técnicas de Mineração de Dados (Han, 2006), passo mais importante no processo de Descoberta de Conhecimento em Banco de Dados (em inglês*, Knowledge Discovery in Data Bases* - KDD) (Fayyad, 1996), são comumente utilizadas no contexto da recomendação. Neste trabalho destacamos as técnicas de Mineração de Dados baseadas em Símbolos.

Mineração de Dados consiste na aplicação de análise de dados e algoritmos que produzem uma relação particular de padrões de dados (Fayyad, 1996). As tarefas de mineração podem ser do tipo (i) Associação, (ii) Classificação, (iii) Clusterização, dentre outros (Fayyad, 1996).

Em Ávila (2010) é realizada recomendação para TV basicamente por meio de técnicas de mineração de dados (regras de associação), assim como em (Kim et *al*., 2010), que utiliza a tarefa de clusterização em uma das etapas do processo de recomendação proposto. Na Seção 2.2 serão discutidos detalhes sobre o processo de KDD, assim como detalhes das tarefas de mineração de dados**.**

b. Técnicas Baseadas em Semântica

Dentre os sistemas de recomendação existem os que recomendam de forma semântica, que utilizam conceitos e técnicas de Representação de Conhecimento (RC) (Russell & Norvig, 2002), como por exemplo, Ontologias (Guarino, 1995), durante o processo de recomendação. RC é a subárea da Inteligência Artificial que avalia como o conhecimento pode ser representado simbolicamente e manipulado de forma automática por máquinas (Russell & Norvig, 2002).

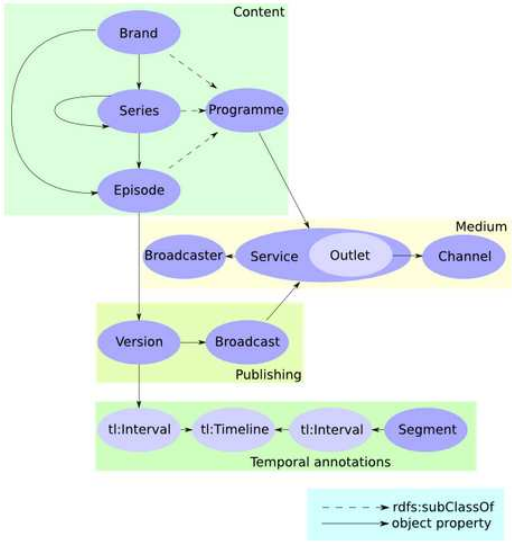
O trabalho de Middleton (Middleton, 2003) realiza um estudo para comprovar o aumento de precisão no processo de recomendação quando se faz o uso da semântica dos dados. Os conceitos de RC podem ser utilizados para diversos fins no

procedimento de recomendação: (i) Integração de dados, (ii) Enriquecimento semântico, (iii) Representação do conhecimento e (iv) Raciocínio automático.

Em Fernández et *al*. (2006) e Kim et *al*. (2011) é realizada recomendação de conteúdo para TV, onde ontologias são utilizadas para estruturar os metadados e possibilitar a inferência de conhecimento. Em Bellekens (2007), o processo de recomendação no contexto da TV, utiliza ontologias para integração de dados de fontes heterogêneas e enriquecimento semântico por meio da Web.

Neste sentindo, está tornando-se padrão a ontologia de programas de TV *BBC Programmes Ontology* (BBC, 2012), amplamente aceita pela comunidade da Web Semântica (Dogdu & Battal, 2010), para descrever conceitos e relacionamentos entre programas. Na Figura 2, é ilustrada parte dos relacionamentos definidos na *BBC Programmes Ontology*.

Figura 2 - *BBC programmes ontology* (BBC, 2012)



II. Técnicas Baseadas nos Modelos Conexionistas

Sistemas baseados no modelo de raciocínio conexionista buscam solucionar um problema usando uma metáfora cerebral para modelagem do sistema (Russell & Norvig, 2002). Dentre as técnicas conexionistas, comumente utiliza-se técnicas baseadas em redes neurais (Russell & Norvig, 2002).

Em (Hsu et *al*., 2007) é realizada recomendação baseada nas informações coletadas e armazenadas no histórico do usuário e combina-as com informações

demográficas e atividades de interesse. O processo é realizado por meio de técnicas de redes neurais artificiais que tentam prever as preferências dos usuários.

III. Técnicas Probabilísticas

Técnicas probabilísticas são utilizadas, por exemplo, para estimar se um usuário gostará ou não de um determinado programa de TV, ou seja, são utilizadas para prever o comportamento do usuário.

Neste sentindo, Ardissono et *al*. (2003) utilizam Redes Bayesianas (Russell & Norvig, 2002) para representar os interesses do usuário e gerar um Guia de Programação Eletrônico (em inglês, Electronic Programming Guide - EPG) personalizado. Truyen & Phung (2007) propõem um modelo probabilístico de preferências que combina a FBC e a FC para recomendação.

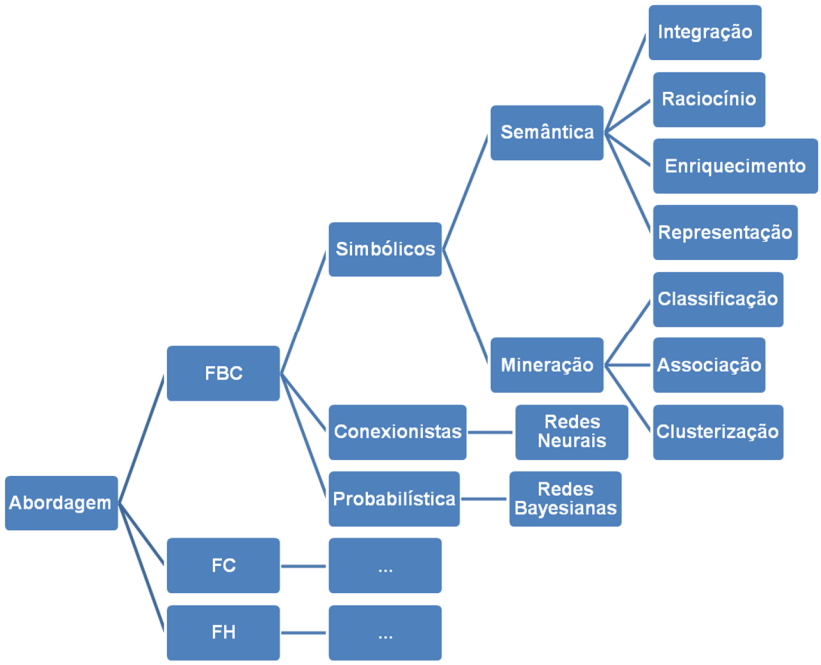
2.1.4.3 Árvore Categórica para Sistemas de Recomendação

Como explanado, sistemas de recomendação podem ser baseados em diversas abordagens e técnicas. Esta característica possibilita um grande número de combinações diferentes a serem investigadas e avaliadas.

Neste trabalho, categorizamos os sistemas de recomendação em 3 (três) níveis de detalhamento (Figura 3). Para cada abordagem, pode-se utilizar técnicas baseadas em modelos simbólicos, modelos conexionistas e/ou probabilísticas (Figura 3).

As técnicas semânticas podem ser categorizadas de acordo com os seus objetivos: (i) Integração de dados, (ii) Enriquecimento semântico, (iii) Representação do conhecimento e/ou (iv) Raciocínio automático. Tal como as técnicas baseadas em mineração que podem ser classificadas de acordo com a tarefa utilizada: (i) Clusterização, (ii) Classificação, (iii) Associação ou (iv)Outras, como ilustrado na Figura 3.

Figura 3 - Árvore categórica para sistemas de recomendação



2.1.5 Avaliação de Sistemas de Recomendação

A avaliação da recomendação de um sistema é uma etapa de extrema importância para a validação das técnicas utilizadas. Essa validação pode ser realizada com um conjunto de usuários reais. Esta abordagem pode ser subjetiva, custosa e onerosa, pois a atividade de escolha destes usuários e a definição do número ideal de indivíduos não são triviais. Por este motivo, também é importante a utilização de métricas objetivas que analisem a precisão das recomendações geradas pelo sistema.

Existem diversas métricas como as listadas em (Montaner, 2003), no entanto, serão destacadas as mais utilizadas na literatura: *Precision*, *Recall* e *Coverage*.

- *Precision*: Mede a porcentagem de itens recomendados que são interessantes para o indivíduo.
- *Recall*: Calcula a porcentagem de itens esperados pelo usuário que são recomendados pelo sistema.
- *Coverage*: Proporção de itens que podem ser recomendados, em relação a todos os itens do sistema.

Neste trabalho analisamos as recomendações geradas por meio da métrica *Precision* (Precisão), com o objetivo de medir a acurácia do processo proposto. A

precisão das recomendações para cada usuário, segue a expressão definida na Figura 4, *P* (Precisão) do processo proposto é a divisão entre as o número de recomendações que o sistema acertou ($R_{Acertadas}$), pelo número de recomendações realizadas pelo processo (R_{Total}).

Figura 4 - Expressão de precisão por usuário

$$P = \frac{R_{Acertadas}}{R_{Total}}$$

A precisão total do processo proposto está definido na expressão da Figura 5. Precisão total do processo proposto neste trabalho (P_T) é a média aritmética das precisões de cada usuário, tal que n é o número de usuários.

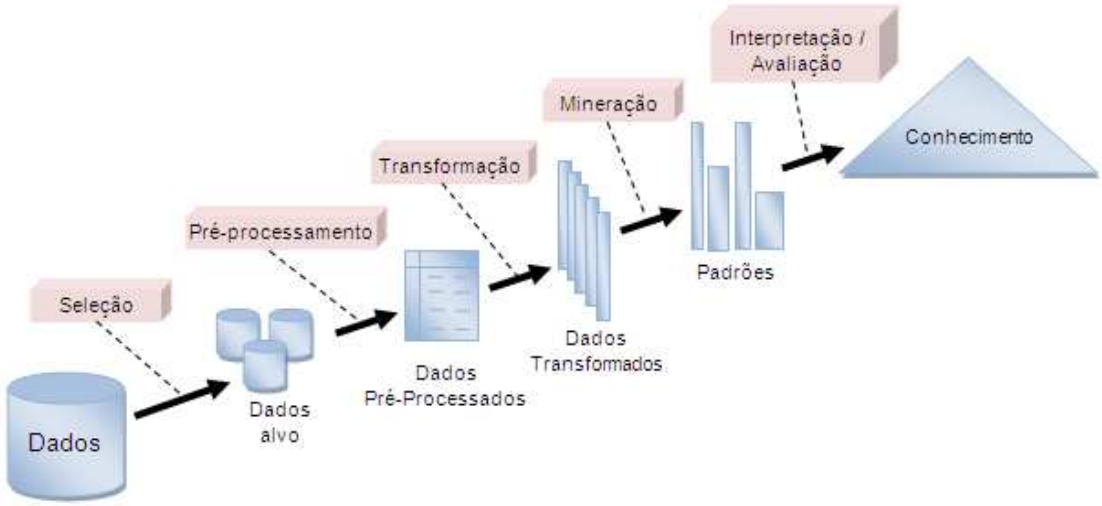
Figura 5 - Expressão da precisão total

$$P_T = \frac{\sum_{i=1}^{i=n} \frac{R_{Acertadas\ i}}{R_{Total\ i}}}{n}$$

2.2 Descoberta de Conhecimento

O processo de descoberta de conhecimento em bancos de dados (em inglês, *Knowledge Discovery in Databases* - *KDD*) é um processo não trivial de identificar em dados padrões que sejam válidos, novos (previamente desconhecidos), potencialmente úteis e compreensíveis, visando melhorar o entendimento de um problema ou um procedimento de tomada de decisão (Fayyad, 1996). Em geral, o processo é interativo, iterativo, cognitivo e exploratório, envolvendo vários passos (Figura 6) com muitas decisões sendo feitas pelo analista (que é um especialista do domínio dos dados, ou um especialista de análise dos dados), conforme descrito:

Figura 6 - Etapas da descoberta de conhecimento



1. Definição do tipo de conhecimento a descobrir, o que pressupõe uma compreensão do domínio da aplicação bem como do tipo de decisão que tal conhecimento pode contribuir para melhorar;
2. Criação de um conjunto de dados alvo (Seleção): selecionar um conjunto de dados ou focalizar um subconjunto onde a descoberta deve ser realizada;
3. Limpeza de dados e pré-processamento: operações básicas tais como remoção de ruídos, quando necessário, e coleta da informação para modelar ou estimar ruído, escolha de estratégias para manipular campos de dados ausentes e formatação de dados de forma a adequá-los à ferramenta de mineração;
4. Redução de dados e projeção (Transformação): localização de características úteis para representar os dados dependendo do objetivo da tarefa, visando a redução do número de variáveis e/ou instâncias a serem consideradas para o conjunto de dados, bem como o enriquecimento semântico das informações;
5. Mineração de dados: selecionar os métodos a serem utilizados para localizar padrões nos dados, seguida da efetiva busca por padrões de interesse numa forma particular de representação ou conjunto de representações; busca pelo melhor ajuste dos parâmetros do algoritmo para a tarefa em questão;
6. Interpretação dos padrões minerados (Avaliação), com um possível retorno aos passos 1-6 para posterior iteração;

7. Implantação do conhecimento descoberto: incorporar este conhecimento ao desempenho do sistema, ou documentá-lo e reportá-lo às partes interessadas.

Neste trabalho daremos foco à etapa de mineração de dados (discutida na subseção 2.2.1) do processo de KDD.

2.2.1 Mineração de Dados

Mineração de Dados consiste na aplicação de análise de dados e algoritmos que produz uma relação particular de padrões de dados (Fayyad, 1996). Esta análise e exploração podem ser realizadas de forma automática ou semiautomática, sobre grandes quantidades de dados, a fim de descobrir regras significativas (Berry & Linoff, 1997).

Desta forma, o principal objeto da Mineração de Dados é detectar relacionamentos e padrões entre os dados, extraindo, assim, o conhecimento que está oculto dentro da base de dados. Os resultados obtidos por meio da Mineração de Dados podem ser utilizados no apoio à tomada de decisão, no gerenciamento de processos, no processamento de pedidos, entre outros contextos (Dias, 2001).

O processo de Mineração de Dados pode ser classificado de acordo com as tarefas que o compõe. As tarefas definem os tipos de algoritmos de Mineração de Dados que poderão ser utilizados.

Este trabalho explora a etapa de mineração de dados com o objetivo de realizar recomendação de conteúdo na TVDi.

2.2.1.1 Tarefas de Mineração de Dados

As tarefas de Mineração de Dados classificam o padrão que se deseja obter da base de dados. Podem ser classificadas como: (i) Regras de associação, (ii) Clusterização, (iii) Sumarização, (iv) Classificação e (v) Regressão (Han, 2006). Na Tabela 1 é realizada uma sumarização das características de tarefas de Mineração de Dados.

Neste trabalho daremos foco à tarefa de Clusterização, a utilizaremos como parte do processo de recomendação proposto. Esta foi escolhida por ser comumente utilizada em sistemas de recomendação baseados em um grande volume de dados, amplamente difundida nos sistemas para Web e por ser considerada de boa precisão para o contexto de recomendação.

Tabela 1 - Tarefas de mineração de dados. Adaptado de (Dias, 2001).

Tarefa	Descrição	Exemplos
Associação	Usada para determinar quais itens tendem a serem adquiridos juntos em uma mesma transação.	• Determinar quais produtos são adquiridos juntos em um supermercado.
Clusterização	Utilizada para agrupar itens de acordo com características que os assemelhem.	• Agrupar usuários de TV por conteúdo visualizado.
Sumarização	Envolve métodos para encontrar uma descrição compacta para um subconjunto de dados.	• Tabular o significado e desvios padrão para todos os itens de dados.
Classificação	Constrói um modelo que possa ser aplicado a dados não classificados a fim de caracterizá-los em classes.	• Identificar a melhor forma de tratamento de um paciente; • Classificar pedidos de crédito.
Regressão	Indicada para estimar o valor de uma variável contínua desconhecida.	• Estimar a renda de uma família; • Estimar o tempo de vida de um paciente.

2.2.1.1.1 Clusterização

Um das tarefas de mineração de dados é a Clusterização (Ochi, 2004), também conhecida por segmentação ou agrupamento, que consiste em dado uma base de dados X, agrupar (clusterizar) os objetos de X de modo que os mais similares sejam alocados no mesmo grupo (cluster) e os menos similares em clusters distintos.

Diversos algoritmos da área de inteligência artificial (aprendizagem de máquina), banco de dados (recursos de manipulação de grandes bases de dados) e estatística (avaliação e validação de resultados), podem ser utilizados para agrupamento de dados. Tais algoritmos são classificados como de aprendizagem não-supervisionada (Russell & Norvig, 2002), por não conhecer a classe de cada amostra do treinamento.

A aprendizagem não-supervisionada baseia-se em observações e descobertas automáticas de informações ocultas, onde se reconhecem padrões por si só. Para alcançar este objetivo, alguns algoritmos de mineração de dados podem ser utilizados.

2.2.1.2 Algoritmos de Mineração de Dados

Existem diversos algoritmos na literatura com foco em agrupamento de dados (Berkhin, 2002). Neste trabalho será dada ênfase à técnica *k-means* (Hartigan & Wong Berkhin, 1979).

2.2.1.2.1 Algoritmo *k-means*

A técnica *k-means* é baseada em particionamento (Berkhin, 2002) e basicamente é caracterizada por uma base de dados com *n* elementos e de um número $k \leq n$, que representa o número de grupos que se deseja gerar. Em suma, o algoritmo pode ser descrito seguinte forma (Hartigan & Wong Berkhin, 1979):

1. Escolhem-se arbitrariamente os *k* objetos do banco de dados que serão os representantes (centroides) de cada um dos *k* clusters;
2. Atribui-se cada um dos demais objetos ao grupo que possua maior similaridade com o centroide;
3. Recalcula-se o centroide de cada grupo, como sendo a média dos objetos atuais do grupo;
4. Repete-se os passos 2 e 3 até que os grupos se estabilizem.

A técnica *k-means* é o algoritmo mais popular de clusterização (Berkhin, 2002), é simples, direto e tem custo linear. Um dos seus maiores problemas é ser sensível à seleção da partição inicial, podendo não convergir para a forma desejada (Jain et *al.*, 1999).

2.2.1.3 Funções de Similaridade

Para realizar o agrupamento de objetos é necessário o uso de funções que calculam a similaridade entre os objetos. No contexto de clusterização existem diversas funções que objetivam calcular a distância entre dois objetos (Jain et. *al*., 1999; Kogan et. *al*., 2005). Para a escolha da função é necessário avaliar os dados e verificar os tipos de dados que serão comparados (Wang, 2006).

Entre os tipos de dados, destacam-se os nominais e categóricos. Os tipos nominais descrevem qualquer objeto, são rótulos, nomes ou códigos que representam valores, e só podem ser comparados se são iguais ou diferentes. Os tipos numéricos

possuem as propriedades dos intervalares além de existir razão entre os seus valores, de tal forma são permitidas operações matemáticas entre os dados.

Baseado no objetivo deste trabalho e no tipo de dado a ser clusterizado, os dados categóricos são transformados em dados numéricos, por meio de pré-processamento de dados, como destacado no Capítulo 5. Neste sentido, foi utilizada a função de Distância Euclidiana. Esta é utilizada, principalmente, para dados numéricos. Desta forma é calculada a distância entre os vetores numéricos, que representam o histórico dos usuários.

A distância euclidiana é expressa segundo a Figura 7. Nesta, *x* e *y* são vetores com *n* atributos numéricos. Assim ($x_a - y_a$) é a diferença entre os atributos de *x* e *y* na posição a. A Distância Euclidiana é a raiz quadrada da soma do quadrado da diferença ($x_a - y_a$), para todo *a*.

Figura 7 - Distância euclidiana

$$E(x,y) = \sqrt{\sum_{a=1}^{n} (x_a - y_a)^2}$$

É importante destacar que a distância euclidiana foi utilizada tanto para a clusterização, segundo o algoritmo k-means, tanto nos métodos de análise de clusters (Subseção 2.2.1.4).

Adicionalmente, foi avaliado o uso da função de *Hamming* (Jain et. *al*., 1999; Kogan et. *al*., 2005) para o processo de clusterização. Esta foi descartada por ser específica para vetores binários. Para o agrupamento de usuários, segundo a definição deste trabalho, não é um vetor adequado, nem tão pouco se adequou aos objetivos principais para a clusterização.

2.2.1.4 Análise de Clusters

Com o objetivo de avaliar a qualidade dos agrupamentos geradas por meio de algoritmos de clusterização diversas técnicas foram propostas na literatura (Tan et *al*., 2006). O objetivo principal destas técnicas é avaliar o quanto objetos de um mesmo grupo são similares entre si e o quanto são diferentes dos demais grupos. O ideal é

que se tenha em um mesmo grupo objetos muito similares entre si e que estes sejam distantes dos objetos dos demais grupos gerados.

O popular método de análise de clusters é o coeficiente de silhueta (*silhouette coeficiente*) (Tan et *al*., 2006). Este combina a coesão com a separação dos grupos. Como especificado na expressão da Figura 8, para cada objeto *i*, calcula-se a distância média deste para todos os outros objetos do mesmo cluster e atribui-se a a_i, em seguida, calcula-se a média da distância do objeto *i* para todos os objetos dos demais clusters e atribui-se a b_i. O valor do coeficiente de silhueta (S_i) é um valor no intervalo [-1,1].

Figura 8 - Coeficiente de silhueta

$$S_i = \frac{(b_i - a_i)}{\max(b_i, a_i)}$$

É importante destacar, que no contexto deste trabalho utilizou-se a média dos coeficientes de silhueta para avaliação dos resultados, como destacado na expressão da Figura 9. Desta forma, foi obtido a média sobre todos os objetos de todos os clusters gerados, obtendo o coeficiente de silhueta (S_t) do processo de mineração utilizado.

Figura 9 - Média coeficiente silhueta

$$S_t = \frac{\sum_{i=1}^{i=n} S_i}{n}$$

Adicionalmente ao coeficiente de silhueta, utilizou-se o cálculo da soma dos erros quadráticos (em inglês, *Sum of the Squared Error* - SSE) (Tan et *al*., 2006). Esta é uma medida que avalia a coesão dos agrupamentos gerados. Basicamente, calcula-se a distância entre cada objeto e o centroide do cluster ao qual pertence. A expressão que define a SSE está definida na Figura 10. Nesta c_i representa o centroide e *x* um objeto pertencente ao cluster c_i.

Figura 10 - Soma dos erros quadráticos

$$\mathrm{SSE} = \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} dist(\mathbf{c}_i, \mathbf{x})^2$$

Neste trabalho, utilizou-se o método de SSE e o coeficiente de silhueta de forma complementar. Na verdade, os métodos indicam de forma aproximada o mesmo valor de k ideal para o algoritmo k-means (Tan et *al*., 2006). Desta forma, o uso conjunto foi utilizado para ratificação da qualidade dos clusters.

Existem outros tipos de métodos que prezam pela qualidade dos agrupamentos (Tan et *al*., 2006). Estes foram escolhidos por: (i) Serem aplicáveis ao algoritmo escolhido (k-means). (ii) Serem não supervisionados.

Na próxima Seção serão discutidos os conceitos da web semântica importantes para este trabalho. Estes são importantes para embasar a estrutura semântica do processo de recomendação deste trabalho.

2.3 Web Semântica

Grande parte dos recursos da Web clássica (também chamada de Web sintática) estão disponíveis em linguagem natural, sem uma padronização que possa ser processada por máquinas (Antoniou & Harmelen, 2008). Esta característica acarreta diversos problemas como: (i) pouca precisão de resultados, muitos resultados irrelevantes são trazidos juntos dos resultados alvo, (ii) nenhum resultado é gerado, quando na verdade se tem itens a serem retornados, (iii) dependência sintática do item, não avaliando a semântica e (iv) resultados são páginas web simples, se a informação estiver separada em vários documentos, a consulta deverá ser realizada manualmente.

Com o objetivo de atenuar problemas da Web clássica e facilitar o entendimento e processamento da informação por máquinas, surgiu o conceito de Web Semântica. Esta pode ser definida como uma extensão da web sintática (Web simplificada, sem semântica para os dados), na qual é dada à informação um significado (Berners-Lee et *al*., 2001), permitindo: (i) processamento automático por

máquinas, por meio da padronização da representação do conhecimento (ii) integração de dados, (iii) reuso de dados e (iv) inferência de conhecimento.

A proposta da Web Semântica é baseada nos conceitos de Representação do Conhecimento (RC), subárea da Inteligência Artificial que avalia como o conhecimento pode ser representado simbolicamente e manipulado de forma automática por máquinas (Russell & Norvig, 2002).

Formalizar a RC permite a interoperabilidade de dados, assim como o raciocínio automático por máquinas, podendo agregar semântica a dados brutos. Acredita-se que toda a web será estruturada formalmente, para fazer uso das vantagens da RC. Para esta formalização, pode-se fazer uso de ontologias (Breitman, 2001).

Para alcançar este enriquecimento semântico são necessárias informações adicionais sobre os dados, são os chamados metadados, dados sobre os dados. Neste contexto, a W3C (*World Wide Web Consortium*) (W3C, 2012) define metadados como informação para a web que pode ser compreendida por máquinas.

Parte deste trabalho se concentra na construção e uso de ontologia, na definição de um modelo de representação de conhecimento capaz de expressar com relevância informações sobre conteúdo multimídia transmitido em sistemas de TVDi, detecção de conexões semântica entre dados e na definição e uso de metadados a serem incorporados no processo de recomendação de conteúdo. Em virtude disso, as subseções seguintes aprofundam mais detalhadamente acerca dos conceitos e construção de ontologias.

2.3.1 Ontologias

A literatura sobre ontologias apresenta distintas definições, com pontos de vista diferentes e até mesmo complementares para uma mesma realidade. Gruber (1993) define ontologia como uma especificação formal e explícita de uma conceitualização compartilhada de conceitos e relacionamentos que podem existir entre agentes ou comunidade de agentes.

Segundo Fesel (2001), em analise a definição de Gruber (1993), *conceitualização* representa um modelo abstrato de algum fenômeno que identifica os conceitos relevantes para ele; *explícita,* significa que os elementos e suas restrições

estão claramente definidos; *formal,* que a ontologia deve ser passível de processamento automático e; *compartilhada*, que reflita a noção de captura de conhecimento concensual, aceito por um grupo de pessoas.

Guarino (1998) define como sendo um modelo abstrato capaz de organizar de forma explícita e formal os conceitos e restrições relacionados a um domínio de interesse. Gómez-Pérez (1999) define ontologia como sendo um conjunto de termos ordenados hierarquicamente para descrever um domínio que pode ser usado como um esqueleto para uma base de conhecimentos. Neste sentido, Gómez-Pérez (2009) caracteriza uma ontologia por meio de sua estrutura, diferentemente das demais definições.

Basicamente ontologia é um modelo, abstrato e formal, que pode ser utilizado para representação do conhecimento, onde seus conceitos são organizados de forma hierárquica e são definidos relacionamentos entre estes.

No contexto de dados multimídia, pode-se descrever a ontologia de um programa ao definir um conjunto de termos representativos. Adicionalmente, as definições associam nomes de entidades do universo (por exemplo, classes, relações, funções, ou outros objetos) com textos legíveis para pessoas, os quais descrevem os nomes que se deseja representar, e axiomas formais que restringem a interpretação e a formação desses nomes (Gruber, 1993).

Este modelo tem sido bastante utilizado nas pesquisas sobre representação do conhecimento para estudos de Inteligência Artificial em diversas áreas, considerando, especificamente, os tipos de entidades que são admitidos em um sistema linguístico (Silva, 2006; Martín et *al.*, 2009; Silva et *al.*, 2009).

O uso de ontologias possui diversas vantagens: (i) interoperabilidade entre sistemas: permitindo reuso, mapeamento de formalismos, compartilhamento de conhecimento, (ii) raciocínio automático e (iii) Descrição semântica de dados.

Neste trabalho ontologias são utilizadas para a estruturação dos conceitos minerados com o objetivo de realizar raciocínio automático sobre a gama de dados minerados. Esta aplicação possibilita o processamento automático por máquinas e facilita o compartilhamento de informações por um grupo de pessoas ou máquinas

(Gruber, 1993). Na próxima subseção serão explanadas as diversas classificações de ontologias, assim como linguagens utilizadas para defini-las.

2.3.1.1 Classificação de Ontologias

Ontologias podem ser classificadas de acordo com diferentes focos: (i) níveis de generalização, (ii) natureza do assunto e (iii) grau de formalidade.

Guarino (1998) realiza classificação de ontologias quanto à generalidade da seguinte forma:

1. Ontologia de alto-nível: descreve conceitos gerais como espaço, tempo e eventos. Conceitos tipicamente independentes, podendo ser facilmente compartilhada por diversos sistemas;
2. Ontologia de tarefa: Descreve o vocabulário relativo a uma tarefa genérica por meio da especialização dos conceitos presentes na ontologia de alto nível;
3. Ontologia de domínio: Descreve o vocabulário especifico de um dado domínio por meio da especialização de conceitos presentes na ontologia de alto nível;
4. Ontologia de aplicação: São ontologias mais específicas, que possuem conceitos relacionados a papéis desempenhados por entidades do domínio no desenvolvimento de alguma tarefa.

Uschold (1996) realiza a classificação de ontologias quanto a natureza do assunto, com o objetivo de definir o propósito da ontologia. Esta classificação seque as seguintes definições:

1. Ontologia de domínio: Descreve domínios particulares, fornecendo vocabulário sobre conceitos, relacionamentos e atividades;
2. Ontologia de tarefas: Descreve um vocabulário de descrição de uma estrutura de resolução de problemas independentemente do domínio onde ocorram;
3. Ontologia de representação: Descrevem conceitos que fundamentam os formalismos de representação do conhecimento.

Ainda na classificação de ontologias Uschold (1996) as categoriza quanto ao grau de formalidade, sendo:

1. Altamente informal: Expressa em linguagem natural;

2. Semiformal: Expressa em uma linguagem natural, estruturada de maneira formal, de forma a atenuar a ambiguidade e aumentar a clareza;
3. Rigorosamente formal: Expressa formalmente em uma linguagem bem definida com teoremas e provas.

2.3.1.2 Linguagens para Construção de Ontologias

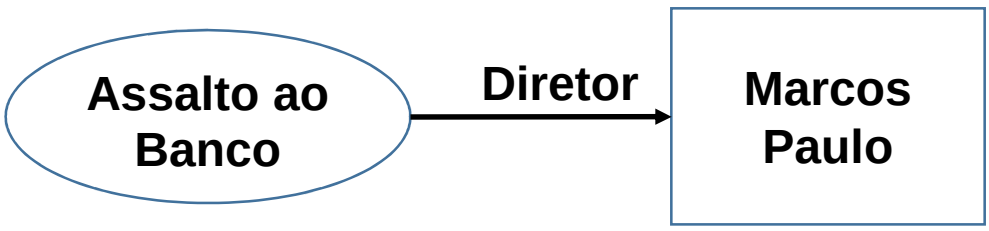
Estão disponíveis diversas linguagens para a construção de ontologias: OIL (Fensel et *al.*, 2000), DAML+OIL (Horrocks et *al.,* 2001), RDF (*Resource Description Language*) (RDF, 2012) e OWL (*Web Ontology Language*) (OWL, 2012), dentre outras. Pelo fato de serem recomendadas pela W3C como linguagem padrão para construção de ontologias, a seguir serão detalhadas as linguagens: (i) RDF e (ii) OWL.

I. RDF

A linguagem RDF foi idealizada pela W3C para padronizar a definição e utilização de matadados. A linguagem fornece um meio universal para expressar informações sobre recursos, além de permitir o intercambio entre aplicações sem a perda de significado. O RDF tenta trazer interoperabilidade ante a multiplicidade de formatos incompatíveis existentes.

O RDF apresenta sintaxe para descrever em 3 (três) conceitos: (i) recursos, que são objetos sobre os quais se quer falar, (ii) propriedades, que descrevem os relacionamentos entre recursos ou suas características e (iii) declarações, que é uma tripla objeto-atributo-valor (ou sujeito-predicado-valor), consistindo de um recurso, uma propriedade e um valor, respectivamente. Uma sentença RDF é apresentada na Figura 11, onde o recurso é o filme “Assalto ao Banco Central”, a propriedade é “diretor” e o valor é “Marcos Paulo”.

Figura 11 - Representação de uma declaração em RDF



II. OWL

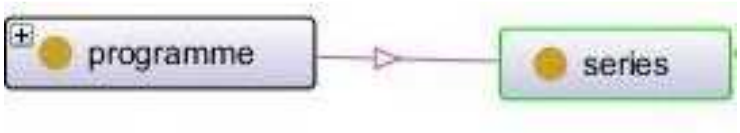
A linguagem OWL foi lançada pelo consócio W3C como padrão para especificação de ontologias, é uma evolução de outra linguagem, a DAML+OIL (Breitman, 2005). É uma linguagem para definição e instanciação de ontologias com o propósito de proporcionar representação de conceitos e seus relacionamentos.

Uma ontologia em OWL pode descrever um domínio, definindo axiomas, classes e propriedades, tal como seus valores e os relacionamentos semânticos existentes entre as entidades. Os principais relacionamentos definidos em OWL são: (i) Gereralização (*superClasseOf*), (ii) Especialização (subClassOf). (iii) Equivalência (EquivalentClass). Na Figura 12 é exemplificada uma declaração em OWL do relacionamento de especialização entre a classe “Programme” e a classe “Series”. Na Figura 13 é apresentado graficamente o relacionamento que indica que “Series” é uma subclasse de “Programme”.

Figura 12 - Declaração de um relacionamento de especialização em OWL

```
<SubClassOf>
        <Class IRI="Series"/>
        <Class IRI="Programme"/>
</SubClassOf>
```

Figura 13 - Representação gráfica de um relacionamento de especialização em OWL



2.3.1.3 Raciocinadores Sobre Ontologias

Raciocinadores são softwares capazes de inferir consequências lógicas a partir de um conjunto de fatos ou axiomas (Russell & Norvig, 2002). Em geral, permitem manipulação de ontologias, suportam tarefa de raciocínio de forma automatizada, possibilitam tarefas de consultas, tal como criação de regras semânticas.

No contexto deste trabalho de mestrado se fez necessário o uso de raciocinadores para inferir os relacionamentos semânticos, que se utilizam das regras semânticas, especificadas e criadas neste trabalho.

Para a escolha do raciocinador ideal a ser utilizado no projeto, foi analisado o estudo comparativo realizado por Kathrin Dentler, entre outros autores (Dentler et *al*., 2011), que avalia diversos raciocinadores considerando características como: (i) Capacidade de inferência, (ii) Código aberto, (iii) Documentação, (iv) Linguagens, (vi) Desempenho e (vi) Usabilidade. A partir desta análise optou-se pelo raciocinador Jena (Jena, 2013), já em uso no projeto KTV.

Jena é um framework Java para construção de aplicações baseadas na Web Semântica. Possui código aberto, permite inferência sobre ontologias no formato OWL e RDF e uma documentação boa para uso e manipulação da API.

2.4 TV Digital Interativa Conectada

A televisão, como outros meios de comunicação, segue a tendência mundial da digitalização de suas plataformas analógicas. O precursor deste processo foi os Estados Unidos na década de 90, espalhando-se em seguida pela Europa, Japão e mais recentemente no Brasil e China. Sendo, assim, uma tendência mundial.

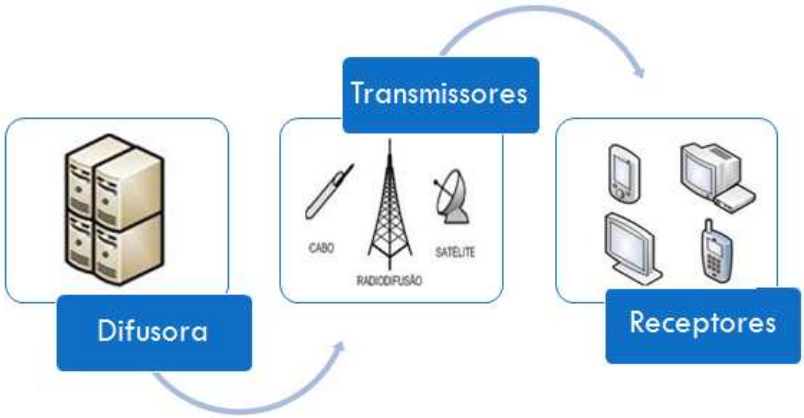
Cada uma das regiões que difundiram a TVD definiu um padrão para seus sistemas digitais: ATSC (*Advanced Television Systems Committee*) (ATSC, 2012) nos Estados Unidos, DVB (*Digital Video Broadcast*) (DVB, 2012) europeu, ISDB (*Integrated Services Digital Broadcasting*) (ISDB, 2012) no Japão, o SBTVD (SBTVD, 2010) brasileiro, abordado na Seção 2.4.1, e o DTMB (*Digital Terrestrial Multimedia Broadcast*) (DTMB, 2012) chinês. Cada um obedecendo as especificidades e objetivos de suas regiões. No entanto, a maioria de sistemas de TVD, mantém um núcleo de convergência, definido pela União Internacional de Telecomunicação (em inglês, *International Telecomunication Union*) – ITU (ITU, 2001), por meio de um modelo de referência para transmissão de sinais de TVD.

De forma positiva, a TV Digital é capaz de distribuir um conteúdo com boa qualidade audiovisual, eliminando problemas típicos em sistemas de TV analógica, como ruídos, distorções nas imagens e fantasmas (Jones et *al*., 2006). No entanto, como nas demais transformações do analógico para o digital, este processo tem sua precisão atenuada.

No geral, um sistema de TV Digital interativa possui ao menos três componentes (Figura 14): (i) Uma estação difusora, responsável pela geração do sinal

digital do conteúdo a ser transmitido. (ii) Um meio de transmissão entre a difusora e os receptores. (iii) Um receptor digital, capaz de receber, decodificar e exibir o conteúdo transmitido.

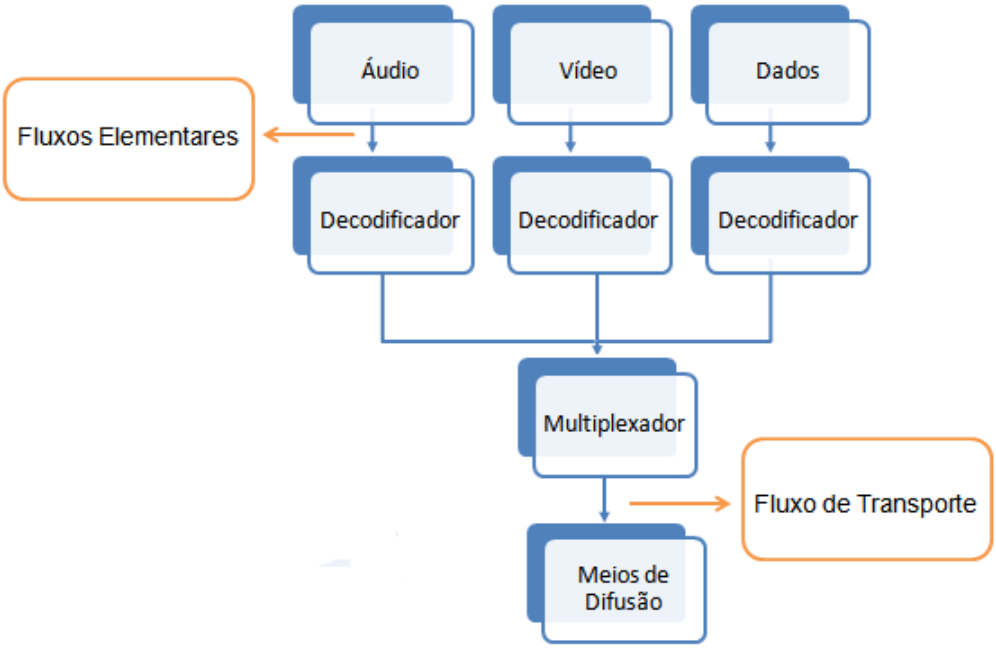
Figura 14 – Componentes básicos do sistema de TV Digital interativa



Adicionalmente aos componentes descritos na Figura 14, o sistema de TV Digital pode possuir estrutura que gera novas possibilidades de interatividade entre os usuários e os provedores de conteúdo, o canal de retorno. Este permite que dados possam ser enviados, via internet, definindo a TV Digital interativa conectada.

Em suma, a geração do sinal digital nas difusoras, de acordo com o modelo de referência da ITU, adotado na maioria dos padrões de TV Digital, permeia as seguintes etapas, ilustradas na Figura 15:

Figura 15 – Etapas da geração de sinal. Adaptado de (Piccioni, 2005).



1. Codificação: Etapa onde é realizada a digitalização e compressão de vídeo, áudio e dados. No contexto deste trabalho, os dados interessantes são os que trazem informações adicionais sobre o conteúdo que está sendo transmitido. Os fluxos codificados são chamados de fluxos elementares;
2. Multiplexação: Responsável por sincronizar as componentes de áudio, vídeo e dados, gerando o fluxo de transporte;
3. Modulação: Etapa onde são definidos os esquemas para transmissão do sinal de acordo com as características específicas de cada sistema de TVDi.

Neste sentido, na próxima Seção serão explanados detalhes do Sistema Brasileiro de TVD, incluindo o *middleware* Ginga (Soares & Lemos, 2007).

2.4.1 Sistema Brasileiro de TV Digital (SBTVD)

O SBTVD foi baseado no sistema japonês de TV Digital (ISDB) e a ele foram incorporadas inovações tecnológicas de compressão, sistemas operacionais e softwares (MONTEZ & BECKER, 2005). Desta forma, surgiu o padrão brasileiro ISDTV-T (*International Standard for Digital Television Terrestrial*). A partir de 2007 passou a ser chamado de ISDB-TB.

Arquiteturalmente, o sinal da TVD é gerado por uma estação difusora, e é transmitido por um meio de difusão até o decodificador, representado por um dispositivo chamado de Set-Top-Box (STB). Por possuir tais funcionalidades, esse dispositivo assemelha-se a um computador com poder limitado de processamento, podendo, inclusive, ser conectado na Web. Torna-se permissivo, assim, o envio e recebimento de dados via STB.

A principal inovação do SBTVD em relação ao ISDB está na inserção de uma camada ao receptor, o *middleware*, responsável por abstrair as particularidades do sistema para aplicações e usuários. No cenário brasileiro, o software responsável por estas atividades é o Ginga (Soares & Lemos, 2007). Este será detalhado a seguir.

2.4.1.1 *Middleware* Ginga

O *middleware* Ginga é a camada de software intermediária, presente no receptor, situada entre as aplicações e o sistema operacional, que oferece uma série de facilidades para o desenvolvimento de conteúdo e aplicativos para TV Digital.

A principal função do Ginga é abstrair as particularidades do sistema para aplicações e usuários, controlando as principais funcionalidades do receptor, inclusive suporte à execução de aplicações interativas e ocultação da complexidade do *hardware* e interfaces de comunicação com sinal digital.

Neste sentido, o Ginga reduz a heterogeneidade para os desenvolvedores de aplicações, fornecendo interfaces de programação padronizadas, tornando as aplicações portáveis nos diversos tipos de STBs (Souza Filho et *al*., 2007). Desta forma, o *middleware* possibilita que conteúdos sejam exibidos nos mais diferentes sistemas de recepção, independente da plataforma de *hardware* do fabricante e tipo de receptor (TV, celular, ...).

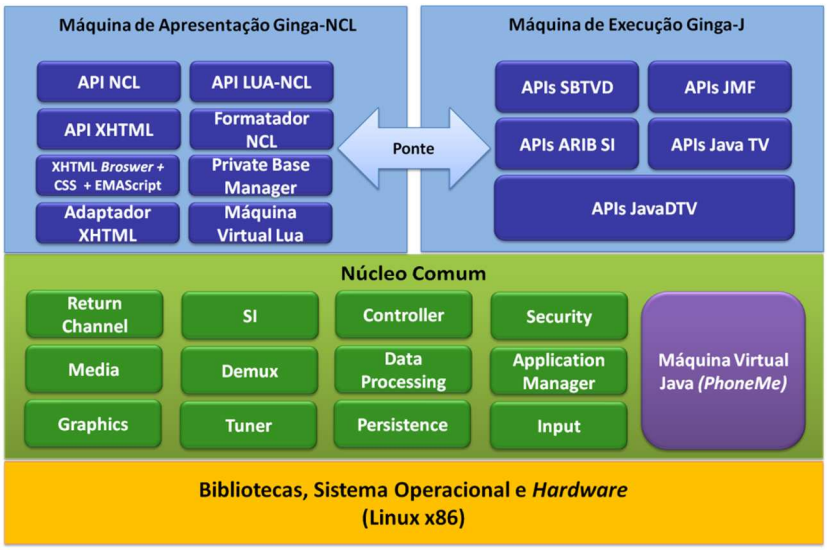
O *middleware* Ginga, é composto por 4 (quatro) macro blocos (Figura 16), tem um ambiente declarativo (Ginga-NCL) baseado na linguagem de programação NCL (*Nested Context Language*) (Soares & Lemos, 2007) e outro imperativo (Ginga-J) (Soares & Lemos, 2007) baseado na linguagem de programação Java. Estes módulos subsidiam o desenvolvimento de aplicações para o ambiente da TVD.

No núcleo comum do *middleware* (Figura 16) são disponibilizadas as funcionalidades comuns para o desenvolvimento de aplicações, no ambiente imperativo e declarativo, para TVD. Dentre os módulos fornecidos, os que possuem maior relação com este trabalho são os seguintes:

- *Tuner*: Responsável pela sintonização dos canais;
- *SI*: Responsável por extrair os metadados transmitidos pelas emissoras.

Os módulos *Tuner* e *SI* possuem extrema conexão com este trabalho por suas funções serem necessárias para a coleta de metadados a serem utilizados no processo de recomendação de conteúdo, proposto neste trabalho, no ambiente da TVDi

Figura 16 - Arquitetura *middleware* Ginga (Araújo, 2011)



Neste trabalho de mestrado são utilizados metadados da TVDi para a realização de recomendação de conteúdo. Desta forma, na próxima Seção serão explanados os metadados no ambiente da TVD.

2.4.2 Metadados em TV Digital

A TVD interativa é caracterizada como sendo um ambiente com uma vasta diversidade de conteúdos multimídia e serviços (Lugmayr et *al*., 2004). Como consequência desta diversidade, a TV passa a enfrentar muitos dos desafios de complexidade e de quantidade de informações enfrentado pelas mídias digitais (Lugmayr et *al*., 2004). Neste cenário surge o uso de metadados (Équille, 2005).

Com uso de metadados é possível tratar serviços e conteúdo digital em uma plataforma de TVD de forma eficiente e com qualidade, viabilizando a compreensão da informação por humanos e máquinas. Basicamente, metadados são dados adicionais que descrevem outros dados, ou seja, descrições concisas dos dados, de forma que estes possam ser usados em momentos de decisões sobre os dados (Équille, 2005).

Atualmente, os principais sistemas de TVD adotam metadados rígidos, destinados a um propósito específico e que não passam por uma padronização de metadados passível de ser estendida (Alves et *al*., 2006). Estes são definidos, principalmente, por meio das Tabelas de Informações de Serviço (SI – *Service Information*) (ABNT, 2007). No entanto, muitos serviços necessitam de informações

mais detalhadas sobre o conteúdo e que não são representadas de maneira satisfatória por tabelas SI. Neste sentido surgiu a necessidade metadados flexíveis, que utilizam estruturas passíveis de customização (Alves et *al*., 2006).

As tabelas SI estendem as tabelas PSI (*Program Specific Information*) do padrão MPEG-2 (MPEG-2, 1992), definindo um conjunto de estruturas que possuem dados descritivos específicos do domínio da TVD. O uso de tais tabelas facilita a criação, tratamento, e a rápida extração de informações (Alves et *al*., 2006).

Como padrão em ascensão de metadados flexíveis pode-se destacar o *TV – Anytime* (TVA, 1999), destacado na próxima subseção.

2.4.2.1 TV - *Anytime*

O TV – *Anytime* (TVA) propõe um padrão público e aberto para o desenvolvimento de sistemas integráveis e interoperáveis possibilitando aos desenvolvedores a manipulação do conteúdo digital aplicável nas fases de pós-produção, distribuição, consumo e interação do conteúdo de TVD interativa (TVA, 1992).

Dentre os objetivos propostos no padrão, segundo Evain e Murret-Labarthe (Evain & Murret-Labarthe, 2003), destacam-se:

- Assegurar que os usuários tenham acesso a conteúdo personalizado, ou seja, de acordo com seus interesses específicos, a partir de uma grande variedade de provedores de conteúdos possíveis;
- Agregar valor ao conteúdo, permitindo ao usuário acessar e utilizar este conteúdo a qualquer momento e onde eles desejarem, sem regras de uso ou restrições de acesso.

Neste padrão há 4 (quatro) categorias de metadados: (i) descrição de conteúdo, (ii) descrição de instância, (iii) usuário e (iv) segmentação (Alves et *al*., 2006):

I. Descrição de conteúdo: Descrição de programas por completo (gêneros de programas, informações de áudio e vídeo);

II. Instâncias: Define informações para dar suporte a mecanismos de localização e anúncio de serviços (no EPG, por exemplo);

III. Descrição de usuário: Determina estruturas para identificação de grupo, perfil e histórico de uso de usuários. Estes metadados possibilitam a criação de aplicações (TVA 1999) como, por exemplo, a personalização e o envio de histórico para provedores;

IV. Segmentação: Descrever fluxos de áudio e vídeo, para permitir o acesso e manipulação em intervalos temporais e de forma aleatória segmentos dos mesmos.

É importante destacar que o TV - *Anytime* não define mecanismo e tecnologias aplicadas no transporte de dados, o que possibilita a sua incorporação em qualquer padrão de TV Digital. Atualmente, existe um esforço para que este padrão seja adotado como modelo de referência no padrão DVB (Alves et *al*., 2006).

2.5 Considerações Finais

Neste Capítulo, foi apresentada uma visão geral da fundamentação teórica deste trabalho, discutindo os principais assuntos relacionados à abordagem proposta. Para isto explanamos as principais características de sistemas de recomendação e a importância destes sistemas no ambiente da TVDi.

Uma vez que este trabalho propõe um processo de recomendação de conteúdo personalizado, por meio de uma abordagem híbrida baseada em Mineração de Dados e nos conceitos da Web Semântica, foi discorrido sobre cada uma destas esferas e realizada a contextualização com a problemática da TVDi, assim como apresentada uma revisão bibliográfica acerca de trabalhos existentes sobre recomendação.

Por fim, foi realizada uma discussão sobre a arquitetura do SBTVD, incluindo o *middleware* Ginga, no qual este trabalho propõe a inserção parcial do módulo de recomendação. No próximo Capítulo é discutida a abordagem deste trabalho para recomendação de conteúdo no ambiente da TVDi conectada inserida no projeto KTV (Lino et *al*., 2011).

Capítulo 3

Recomendação na Plataforma KTV

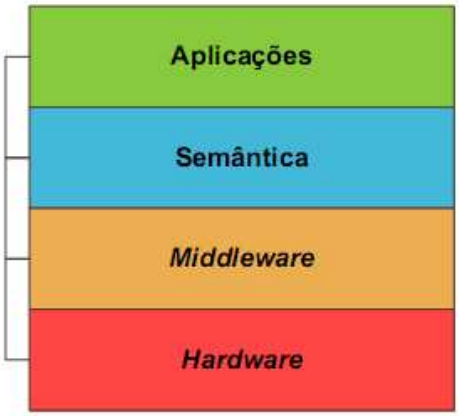
Neste Capítulo é apresentada a abordagem para recomendação de conteúdo utilizada neste trabalho de mestrado. Inicialmente será apresentada a contextualização com o projeto *KnowledgeTV*. Em seguida é apresentado o processo geral para recomendação de conteúdo adotado. Por fim, é introduzido o módulo proposto e utilizado para recomendação de conteúdo multimídia.

3.1 Contextualização com o Projeto *Knowledge TV*

O projeto *Knowledge TV* (KTV) (Lino *et al*., 2011) faz parte do programa Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para a Informação e Comunicação (CTIC), gerenciado pela Rede Nacional de Ensino e Pesquisa (RNP), formado por diversas universidades brasileiras, dentre as quais a Universidade Federal da Paraíba (UFPB), promovendo o intercâmbio de conhecimento na rede.

Dentre outros aspectos, o projeto KTV propõe uma camada semântica, baseada nos conceitos da Web Semântica, para prover serviços na plataforma da TV Digital, tais como: consulta semântica e recomendação de conteúdo. Desta forma, a arquitetura da TVDi, segundo a proposta do projeto KTV, é dividida em quatro camadas, como indicado na Figura 17: (i) Aplicações podem utilizar serviços fornecidos na camada semântica. (ii) Semântica provê serviços semânticos, com estruturação e modelagem de dados. (iii) O *Middleware* é responsável por abstrair detalhes dos dispositivos de *hardware*, facilitando a comunicação entre o *hardware* e as camadas superiores. (iv) A camada de *hardware* é composta por todos os componentes físicos que operam em um ambiente de TVDi.

Figura 17 - Arquitetura conceitual do projeto KTV

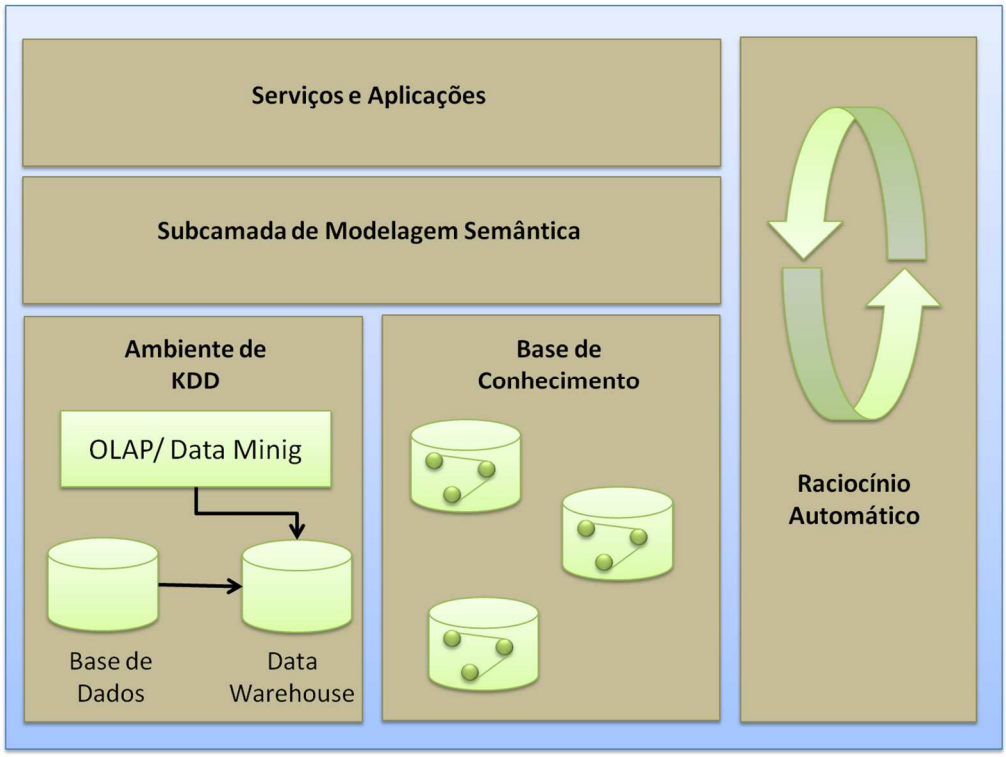


É importante ressaltar que as camadas arquiteturais do KTV (Figura 17) podem se comunicar além das camadas subjacentes. Outro detalhe é que parte da camada semântica, desenvolvida no KTV, está inserida no *middleware*. Esta introdução foi necessária para que fosse possível o fornecimento de recomendação de conteúdo e consulta semântica.

Na arquitetura conceitual do projeto KTV, estão previstas as seguintes entidades (Figura 18):

- Ambiente de KDD para TV Digital: Relaciona tecnologias de um ambiente de Descoberta de Conhecimento em Base de Dados (Han, 2006);
- Bases de conhecimento: Refere-se aos casos reais de armazenamento de dados, modelados de acordo com uma abordagem de modelagem semântica, especificado na subcamada de modelagem semântica (Figura 18);
- Subcamada de modelagem semântica: Encapsula o conhecimento por meio de métodos e linguagens formais que permitem processamento automático por agentes computacionais;
- Raciocínio Automático: Responsável por todas as operações de raciocínio automático efetuadas sobre a subcamada de modelagem semântica;
- Serviços e Aplicações: Fornece serviços e aplicações baseadas em representação de conhecimento, raciocínio automático e KDD.

Figura 18 - Arquitetura conceitual da camada semântica



A partir da arquitetura geral da plataforma KTV diversos trabalhos foram e estão sendo realizados com objetivo de concretização do projeto, tais como: (i) Estudos e desenvolvimento de agentes no *middleware*, permitindo a coleta de metadados dos conteúdos. (ii) Elaboração de modelagem e descrição para os dados analíticos, semânticos e operacionais da plataforma KTV. (iii) Análise, especificação e geração de ontologias para representação de conhecimento e raciocínio automático. (iv) Especificação, desenvolvimento e análise de mecanismos consultas semânticas. (v) Especificação, desenvolvimento e análise de serviços de recomendação semântica.

A proposta deste trabalho instancia a arquitetura da Figura 18 para o serviço de recomendação semântica de conteúdo. Desta forma, a recomendação está relacionada à especificação de bases de conhecimento, utilização do ambiente de KDD (Figura 18), mecanismo para raciocínio automático e fornecimento de um serviço de recomendação. A abordagem proposta para o desenvolvimento de tais entidades será apresentada no próximo capítulo.

A seguir, discute-se a abordagem de recomendação de conteúdo proposta neste trabalho e inserida no contexto da plataforma KTV.

3.2 Abordagem para Recomendação de Conteúdo

Neste trabalho, propomos uma abordagem híbrida para recomendação de conteúdo. Esta abordagem foi escolhida por 3 (três) motivos: (i) Validar o objetivo específico de permitir recomendação multimídia em TVDi conectada, considerando, assim, os dois cenários definidos na literatura (Adomavicius & Tuzhilin, 2005), FC e FBC. (ii) Validar a arquitetura genérica de recomendação de conteúdo proposta neste trabalho, garantindo que esta pode ser instanciada para ambos os tipos de recomendação de conteúdo. (iii) Considerar o desafio de recomendação de conteúdo multimídia no ambiente da TVDi, como sendo um ambiente de experiência multiusuário, permitindo a atenuação dos problemas das abordagens em separado.

É neste contexto que serão utilizadas técnicas para Filtragem Baseada em Conteúdo (FBC) e para Filtragem Colaborativa (FC). É importante destacar que este trabalho adapta o processo Híbrido de recomendação definido por Fernández, entre outros autores, (2006). Esta escolha foi realizada por se tratar do mesmo cenário de TVDi considerado neste trabalho de mestrado.

Neste sentido, a principal contribuição deste trabalho no que diz respeito às abordagens anteriores é a capacidade de inferir novos conhecimentos a partir da semântica do conteúdo de TV, tal como a integração da Mineração de Dados neste processo. Todo este processo será detalhado nas subseções a seguir.

3.2.1 Abordagem para FC

Na estratégia com FC é realizada a troca de experiências entre usuários. São detectados grupos de usuários com comportamentos semelhantes, e a recomendação de conteúdo é realizada entre estes, o que possibilita recomendações inesperadas e novas ao usuário.

Basicamente, a tarefa de clusterização é realizada sobre os dados dos históricos dos usuários. É detectado qual o usuário x do cluster possui maior similaridade com o usuário y, para o qual o serviço de recomendação será prestado. Um programa aleatório visualizado por x e não por y, que está sendo transmitido ou próximo de ser transmitido, é recomendado pra o usuário y.

Como forma de aperfeiçoar a decisão de qual programa recomendar para y, a partir de x, regras semânticas são utilizadas, destacadas no Capítulo 4, Seção 4.3.4.1. As regras detectam qual programa da grade (semanal ou mensal) é mais similar semanticamente com os programas recomendados por x à y. Caso não exista nenhum item que satisfaça as condições, um item aleatório será recomendado.

Com o objetivo de possibilitar o processo de recomendação descrito, foi especificada uma arquitetura e avaliada, por meio da instanciação da arquitetura, módulos capazes de prover o serviço de recomendação de conteúdo no âmbito da plataforma KTV, o *Recommender Knowledge* TV (RKTV), introduzido na próxima Seção.

3.2.2 Abordagem para FBC

Basicamente, no contexto deste trabalho e em sistemas de recomendação em geral, existem dois tipos de usuários (STBs): (i) STBs novos, onde inexiste histórico de uso. (ii) STBs com histórico de uso. Para cada um dos tipos de usuários o processo de recomendação comporta-se de forma diferente.

3.2.2.1 Abordagem para FBC: STB sem histórico de uso

Para situações em que o usuário é novo e não existe nenhum item a ser recomendado, pois este não possui histórico e o sistema não tem informações prévias sobre o usuário, o procedimento segue da seguinte forma: no momento em que o usuário solicita a recomendação, é verificado qual o programa que está sendo visualizado. Regras semânticas são aplicadas sobre os dados provindos das emissoras, indicando a programação futura (semanal ou mensal), possibilitando raciocínios sobre gêneros de programas, por exemplo. O uso de regras semânticas é importante para que programas que não possuem apenas aproximações sintáticas (de grafia) possam ser recomendados ao usuário. Esta mesma estratégia é utilizada por Fernández, entre outros autores, (2006), Kim, entre outros autores, (2010) e Aroyo, entre outros autores, (2007).

Esta estratégia de Filtragem Baseada em Conteúdo possibilita recomendações mais próximas do esperado pelo o usuário, dado que a similaridade de programas, provavelmente indicará algo do mesmo gênero ou próximo do programa atual.

3.2.2.2 Abordagem para FBC: Usuário com histórico de uso

A recomendação por FBC no caso em que o STB possui um histórico de uso se comporta da seguinte forma: é realizada a FC, já destacada neste trabalho, e como refinamento são aplicadas regras semânticas, destacadas no Capítulo 4, Seção 4.3.4.1. Este raciocínio permite detectar qual conteúdo alvo (programação futura) é mais similar semanticamente ao histórico do STB.

Considerando esta abordagem, incialmente verifica-se usuários que possuem o mesmo perfil do usuário U_i, que solicitou o serviço. Nesta etapa é realizada clusterização por meio dos perfis dos usuários. Os usuários com perfis mais similares com U_i pertencem ao mesmo cluster (*C*) de U_i.

Com o objetivo de refinar a recomendação, as regras semânticas são aplicadas dentro do cluster *C*, entre os conteúdos a serem recomendados (pelos vizinhos de U_i, do cluster *C*) e os conteúdos do histórico de U_i. Aqueles que possuem maior similaridade com os conteúdos vistos por U_i, serão recomendados.

3.3 Módulo *Recommender Knowledge TV* – RKTV

Assim como na web, ambiente com grande diversidade de conteúdos; com o advento da TVDi o usuário começou a interagir com um ambiente com disponibilidade numerosa de canais e serviços. Neste sentido, personalizar o conteúdo da TV de acordo com os interesses dos usuários tornou-se uma tarefa relevante. É neste contexto que sistemas de recomendação, comumente utilizados na web, emergem como solução possível para auxiliar a escolha de conteúdo no ambiente da TVDi.

O ambiente da TV tem uma série de características que torna o ambiente desafiador para sistemas de recomendação de conteúdo: (i) Monousuário, onde um único indivíduo faz uso do sistema, ou multiusuário, onde grupos de pessoas se reúnem para consumir o conteúdo disponibilizado, (ii) Multiplataforma e (iii) Habitualmente usuários de TV são passivos e utilizam o controle remoto como forma de entrada de dados. Estas características diferem o ambiente web, onde a recomendação já é bastante difundida, do ambiente da TV. Estas peculiaridades estão sendo levadas em consideração neste trabalho, principalmente por ser proposta uma recomendação de conteúdo pelo uso do STB, e não pela identificação precisa do usuário, e por coleta implícita de dados.

O objetivo geral desse trabalho é propor uma abordagem de recomendação adequado à arquitetura da TV Digital interativa (TVDi) brasileira, considerando-a como um ambiente de convergência (TV e Web) (RESNICK & VARIAN, 1997), permitindo o direcionamento de conteúdo de acordo com o uso do STB. O trabalho propõe uma abordagem baseada em técnicas de mineração de dados integradas a conceitos semânticos.

A semântica é integrada neste trabalho para a representação do conhecimento minerado e apoio ao processo de recomendação. Esta representação permite uma formalização do conhecimento adquirido, assim como a interoperabilidade destes dados e o raciocínio automático sobre os dados a serem recomendados.

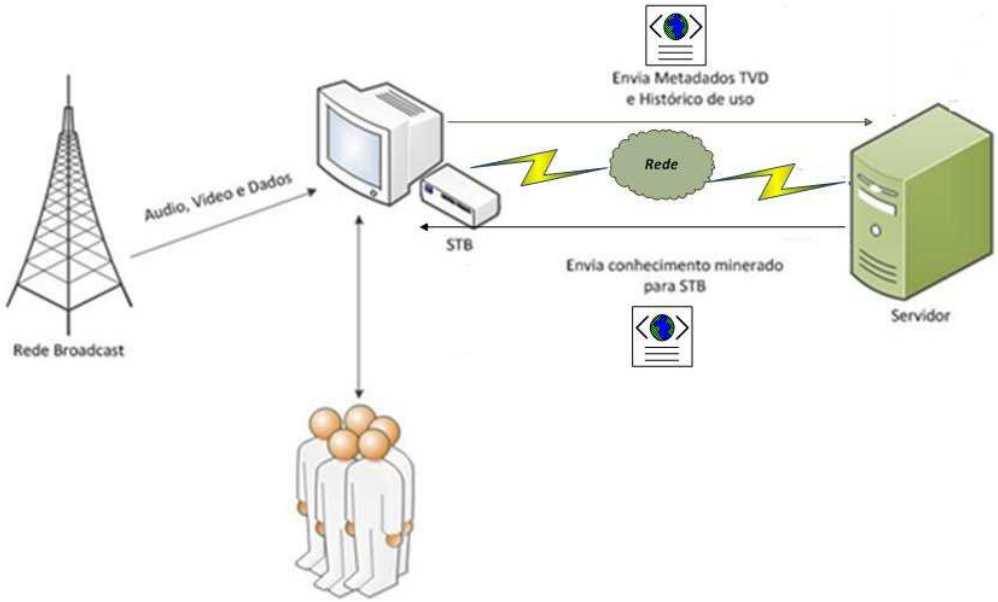
Para possibilitar a recomendação de conteúdo no SBTVD, este trabalho propõe o desenvolvimento do módulo *Recommender Knowledge TV* (RKTV) observando os seguintes requisitos:

- Ser genérico, possibilitando o serviço de recomendação independente do ambiente, podendo ser adaptando para outros sistemas de TV, assim como na web;
- Prover descoberta de conhecimento para o usuário final, provedor de conteúdo, assim como para agente de *marketing*;
- Disponibilizar o serviço de recomendação de conteúdo após o processo proposto;
- Abstrair o complexo processo de recomendação de conteúdo por mineração de dados, que requer uma plataforma física de grande desempenho, para quem os solicite;

A Figura 19 apresenta uma visão de alto nível do funcionamento básico do processo de recomendação na plataforma KTV, especificado neste trabalho de mestrado. A rede de transmissão de TVDi envia o áudio, vídeo e os dados para o *Set-Top-Box* (STB). O usuário interage com a TV por meio de aplicações. O STB envia os dados do usuário via arquivos XML (eXtensible Markup Language) (XML, 2012) por meio do canal de retorno para o servidor KTV. Este arquivo é construído no provedor por meio dos agentes provedor e monitor na plataforma KTV. No servidor KTV técnicas de recomendação são aplicadas, baseadas em mineração de dados e

conceitos da Web Semântica. A recomendação é enviada via Web, quando solicitado o serviço de recomendação.

Figura 19 – Visão geral do processo de recomendação na plataforma KTV



Para alcançar os objetivos do trabalho, seguindo a visão geral da Figura 19 e os requisitos, foi proposta uma arquitetura baseada em módulos a ser destacada no Capítulo 4.

3.4 Considerações Finais

Neste Capítulo foi apresentada uma visão geral do serviço de recomendação de conteúdo no projeto KTV, no qual este trabalho de mestrado está inserido. Inicialmente foram explanadas as principais características da plataforma KTV, definidas no projeto KTV.

Em seguida foram detalhadas as abordagens de recomendação de conteúdo utilizadas neste trabalho de mestrado, uma abordagem híbrida baseada em tarefas de mineração de dados e em conceitos da Web Semântica.

Por fim foi introduzida a arquitetura do módulo RKTV, foco deste trabalho, dando uma visão geral do processo de recomendação de conteúdo na plataforma do KTV por meio do RKTV. No próximo Capítulo será detalhada toda a arquitetura RKTV, tal como as ontologias e as regras semânticas propostas neste trabalho.

Capítulo 4

Recommender Knowledge TV

Neste Capítulo é detalhado o módulo *Recommender Knowledge TV* (RKTV), explanando o processo de recomendação de conteúdo em TV Digital interativa conectada, baseado em técnicas de Clusterização e conceitos da Web Semântica. Inicialmente será apresentada uma visão geral da arquitetura proposta, seguida de uma visão detalhada sobre cada módulo da abordagem e uma proposta de integração arquitetural ao Núcleo Comum do Ginga (GingaCC – Ginga Common Core).

4.1 Arquitetura RKTV

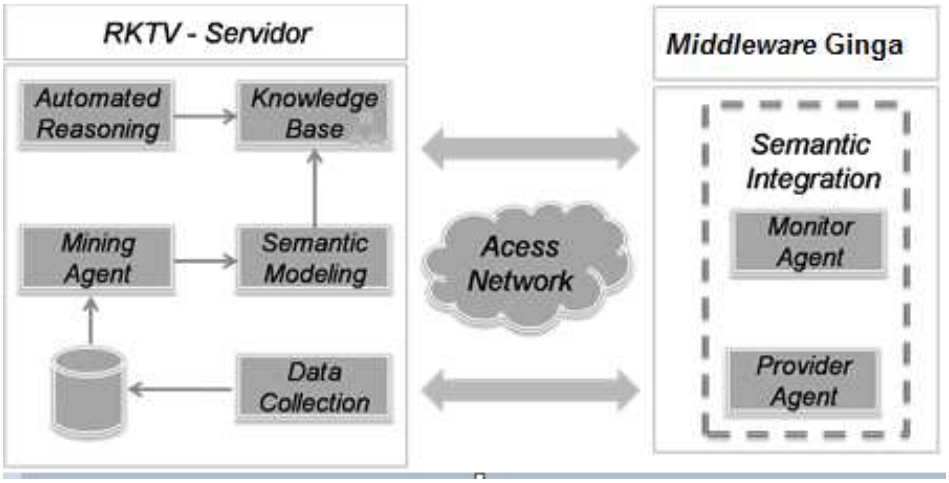
Sob a perspectiva da execução das operações previstas neste trabalho, tais como tarefas e técnicas de mineração de dados para recomendação de conteúdo, considerou-se o ambiente de TVDi conectada brasileiro, considerando o *middleware* Ginga e a capacidade de processamento dos decodificadores de sinais, os STB. Neste sentido, a conectividade foi necessária por considerarmos a capacidade limitada de processamento de dispositivos de STB, tal como a capacidade reduzida de armazenamento. A conectividade permite a consecução de uma arquitetura cliente servidor. É importante destacar que este trabalho de mestrado está inserido em um projeto que objetiva a especificação e implementação de serviços semânticos para o ambiente de TVDi brasileiro.

A Figura 20 explana a arquitetura cliente-servidor proposta neste trabalho de mestrado. Nesta existe um componente como extensão do *middleware* Ginga, o *Semantic Integration*, inserido no núcleo comum do *middleware*, e outro módulo localizado no servidor, no qual é concentrada a maior parte do processamento de informações para dar suporte ao serviço de recomendação de conteúdo.

O uso do serviço de recomendação do RKTV se dá via interface (API – Application Program Interface), onde o cliente (agente que necessita da recomendação) fornece seus dados e o RKTV fornece a recomendação personalizada. Para este trabalho, foram consideradas recomendações de conteúdo multimídia, com a construção, inclusive, de uma ontologia para recomendação de conteúdo.

É importante destacar que este trabalho de mestrado foi viabilizado, analisado e especificado considerando dados multimídia do domínio de filmes, no entanto qualquer ambiente que disponibilize dados e necessite de recomendação de conteúdo multimídia, pode solicitar o serviço via Web para o servidor RKTV, tal como o ambiente de TVDi. Neste sentido, a arquitetura do RKTV considera uma extensão do *middleware* Ginga para a coleta de dados *broadband* e *broadcast*.

Figura 20 - Visão geral do RKTV



A abordagem proposta é estruturada em 8 (oito) módulos, como destacado na Figura 20, distribuídos em 2 (dois) componentes: (i) O de extensão do *Common Core* do *middleware* Ginga e (ii) O servidor, local onde será realizado o processamento para a disponibilização do serviço de recomendação. Cada módulo possui funcionalidades especificas, como destacado a seguir:

- *Semantic Integration:* Responsável por estruturar, gerenciar e enviar para o servidor, as informações da programação visualizada no STB (atividade realizada no *Provider Agent* em conjunto com o *Monitor Agent*). Este componente é uma extensão do *Common Core* do Ginga;

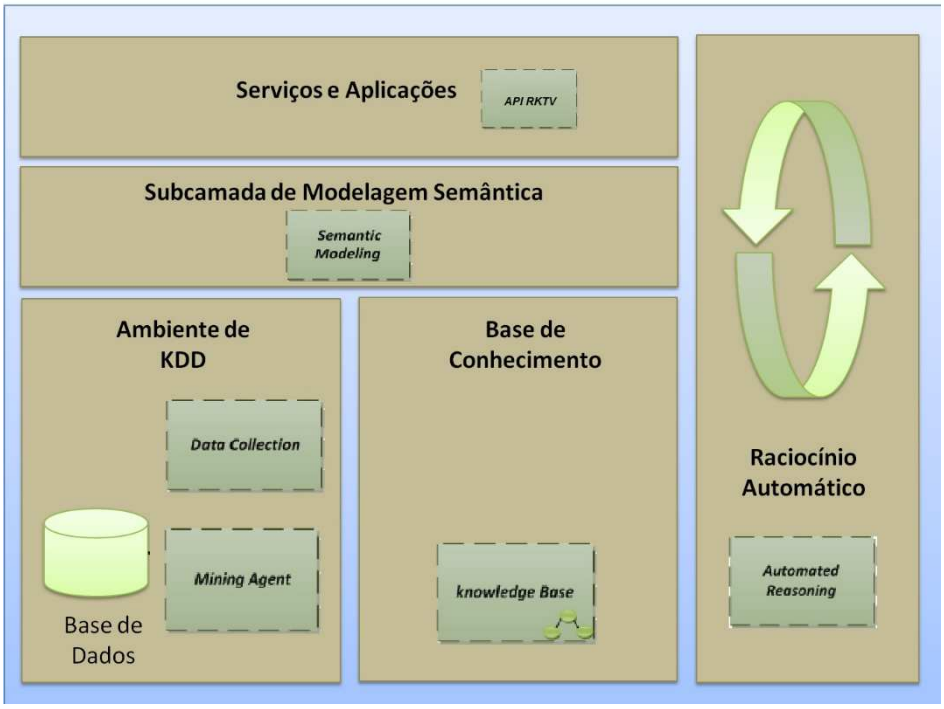
- *Monitor Agent*: Monitorar o comportamento dos usuários do STB em relação ao conteúdo exibido na TVDi conectada;
- *Provider Agent*: Tem a função de capturar as informações sobre a programação visualizada pelo o usuário;
- *Data Collection:* Capturar os dados que serão processados no componente servidor RKTV;
- *Mining Agent:* Responsável por todo o processo de mineração de dados sobre os dados coletados do STB e da emissora;
- *Semantic Modeling*: Tornar homogenia a descrição do conteúdo multimídia minerado, estruturado por meio de ontologias, tal como permite o raciocínio automático sobre o conteúdo;
- *Knowledge Base*: Armazenar todo o conhecimento gerado no processo proposto em forma de ontologias. Como instâncias da ontologia proposta neste trabalho de mestrado, foi utilizada a base de dados NetFlix (NetFlix, 2013). Esta foi minerada e transcrita para a ontologia segundo o procedimento descrito na Seção 4.3.3;
- *Automated Reasoning*: Responsável por processar as informações presentes nas bases de conhecimento, gerando recomendações de conteúdo compatíveis aos interesses do usuário.

Os módulos do componente STB, apresentado na Figura 20, são integrados ao Núcleo Comum do *middleware* Ginga onde são definidas funções básicas de sistemas de TVD, tais como: exibição e controle de mídias, controle de recursos do sistema, canal de retorno, dispositivos de armazenamento, acesso a informações de serviço e sintonização de canais (Soares & Lemos, 2007). Esta extensão foi necessária para facilitar a possibilidade de acesso direto às informações de uso do STB e possibilitasse o serviço de recomendação de conteúdo especializado por aparelho.

Em linhas gerais, o RKTV instância a arquitetura conceitual do KTV (Figura 18). Neste sentido, cada um dos módulos propostos neste trabalho objetivam oferecer um serviço baseado em semântica para o ambiente da TVDi. A Figura 21 relaciona a arquitetura proposta neste trabalho com a arquitetura conceitual do KTV. Nas

próximas subseções serão abordados os detalhes de cada um dos componentes e seus módulos.

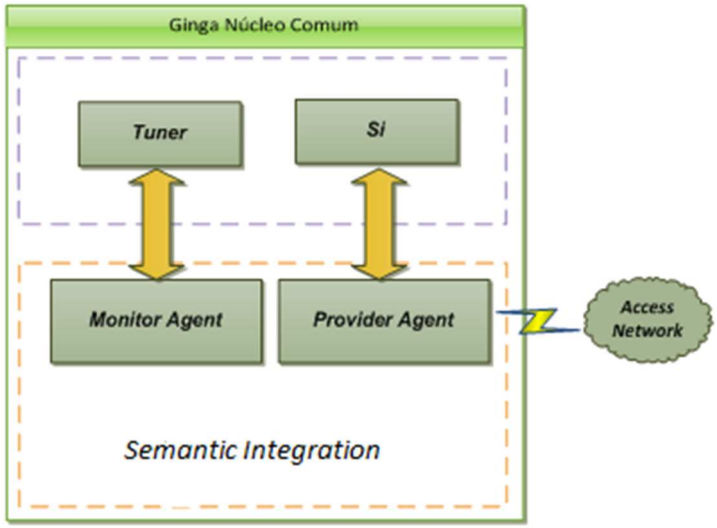
Figura 21 - Arquitetura geral do RKTV integrada ao KTV



4.2 Extensão do Núcleo Comum do Ginga

Nesta Seção serão detalhados os módulos do componente de extensão CC do RKTV, considerando um cliente de TV. Na Figura 22 são destacadas as interações entre os módulos do componente cliente. É importante destacar que estes módulos foram desenvolvidos por outros integrantes do projeto KTV e que o serviço de recomendação de conteúdo recebe os dados monitorados e coletados por tais agentes.

Figura 22 – Arquitetura Extensão CC KTV



4.2.1 *Monitor Agent*

O *Monitor Agent* tem como função principal monitorar o comportamento dos usuários do STB em relação ao conteúdo exibido na TVDi conectada. Este monitoramento extrai o canal que o STB está sintonizado, tal como o horário inicial em que foi sintonizado. Estas informações devem ser enviadas ao módulo *Provider Agent,* que tem comunicação com o módulo *SI.*

O *Monitor Agent*, para monitorar o comportamento dos usuários do STB, está em comunicação constante com o módulo *Tuner* (Soares & Lemos, 2007), responsável pela sintonização dos canais, do *middleware Ginga.* Estas interações são ilustradas na Figura 22.

4.2.2 *Provider Agent*

O *Provider Agent* tem como função principal capturar os metadados da programação visualizada pelo o usuário. Este módulo recebe os dados de interações do usuário por meio do módulo *Monitor Agent* e captura os metadados via Web ou módulo SI, responsável por extrair os metadados transmitidos pelas emissoras, do *middleware Ginga* (Soares & Lemos, 2007) (Figura 22).

A captura de metadados via Web foi realizada por busca em sites especializados em divulgar a programação dos canais. No âmbito deste trabalho, foram coletados metadados de categorias de filmes. Esta abordagem é necessária por não se ter sempre todos os metadados necessários por *broadcast* e por, muitas

vezes, o enriquecimento semântico aprimorar técnicas de recomendação de conteúdo (Aroyo et. *al*, 2006).

Para o objetivo principal do trabalho alguns dados mínimos devem ser coletados a partir do *Provider Agent*. Estes deverão ser utilizados nos demais módulos especificados, auxiliando o serviço de recomendação de conteúdo. Para isto, foram adaptados os metadados definidos em (Araújo, 2011), reduzindo seu número e adequando-os ao contexto deste trabalho. Na Tabela 2 são listados os metadados a serem obtidas e suas respectivas funções.

Tabela 2- Dados extraídos a partir do *Monitor Agent*

Dado	Função
STB_id	Identificar o STB
Program_name	Identificar nome do programa
Channel	Identificar o canal da TV
Program_genre	Gênero do programa
Start_time	Data e horário em que o usuário começou a assistir ao programa
End_time	Data e horário em que o usuário terminou de assistir ao programa

4.2.3 *Semantic Integration*

O módulo tem como função estruturar e gerenciar em um arquivo XML as informações do histórico de uso do STB. Tais dados são obtidos por meio do *Monitor Agent* e do *Monitor Provider* (Figura 22), recuperando, assim, *o* comportamento do usuário.

Desta forma será realizada uma coleta implícita de dados, na qual as interações com o conteúdo são capturadas no *Monitor Agent e Provider Agent* e estruturadas no *Semantic Integration* (Figura 22). Neste sentido, a recomendação será orientada ao padrão de uso do STB e o sistema aprenderá o tipo de programação visto por meio do STB.

O arquivo gerenciado no módulo *Semantic Integration* conterá os metadados definidos na Tabela 2. Estes serão enviados para o módulo *Data Collection*, do componente servidor, via canal de retorno (Figura 20), como detalhado na Seção 4.3.1.

4.3 Componente Servidor

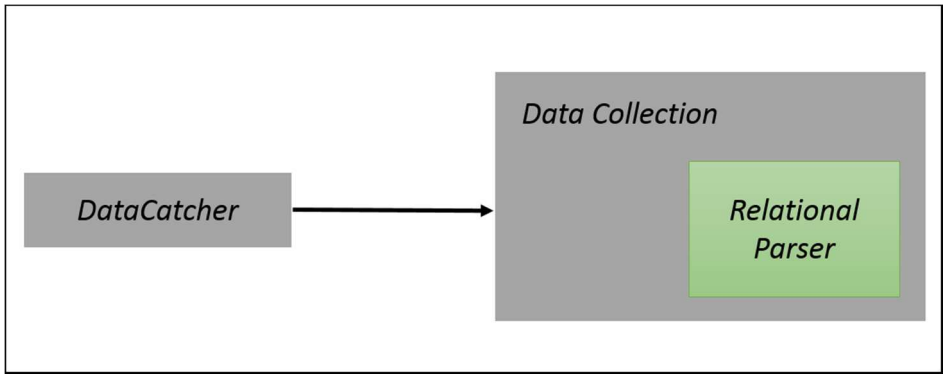
Nesta Seção serão detalhados os módulos do componente servidor do RKTV. É importante destacar que estes módulos instanciam a arquitetura geral da plataforma KTV.

4.3.1 *Data Collection*

O Módulo *Data Collection* tem como função principal receber os dados do módulo *DataCatcher* (Araújo, 2011), responsável por receber as informações obtidas no middleware, através dos agentes Monitor e Provedor. Estes são enviados por meio do Módulo *Semantic Integration* e representam o histórico de uso dos dispositivos STB (Figura 20). O envio é realizado via canal de retorno, em formato XML.

Os dados capturados pelo *DataCatcher*, na arquitetura no KTV, são enviados para o *Data Collection* (Listados na Tabela 2). Neste, o *Relation Parser* (Figura 23) é responsável por depositar em um banco de dados relacional que é submetido ao processo de mineração de dados (Figura 20). Neste sentido, o *Data Collection* transforma os dados operacionais do componente cliente que são utilizados no processo de recomendação de conteúdo.

Figura 23 - Detalhamento *Data Collection*



É importante destacar que não é escopo deste trabalho de mestrado, o levantamento de requisitos de privacidade e segurança para os dados dos usuários. Estas atividades são destacadas como trabalhos futuros no contexto da plataforma KTV.

Adicionalmente o módulo *Data Collection* busca as programações das emissoras na web (semanais ou mensais). Esta funcionalidade foi necessária para que fosse possível a recomendação de conteúdo de acordo com a grade disponível e atualizada da TVDi. Desta forma, com o comportamento no passado do usuário, espera-se recomendar algo de seu interesse na grade presente ou futura.

4.3.2 *Mining Agent*

Responsável por todo o processo de mineração de dados sobre as fontes de dados. O Módulo *Mining Agent* (Figura 20) aplica a tarefa de clusterização (ou agrupamento) (Ochi, 2004) sobre os dados coletados.

É importante destacar que apesar da plataforma KTV prever modelos de dados para dados analíticos, modelo estrela (Kimball, 1998) definido na dissertação de um dos integrantes do projeto (Patrício Júnior, 2012), este trabalho de mestrado não abordará esta modelagem, dando foco diretamente no processo de mineração de dados sobre os dados operacionais.

Neste trabalho é dado foco à clusterização por ser comumente utilizada para recomendação de conteúdo (Kim et *al,* 2011). No entanto, nosso objetivo não é analisar e propor algoritmos de clusterização, mas apenas mostrar a viabilidade da aplicação desta tarefa no ambiente da TVDi. Desta forma, propomos o uso da técnica *k-means* (detalhada na Seção 2.2.1.2.1), utilizando similaridade por distância euclidiana, por ser um dos algoritmos mais simples e populares de agrupamento (Hartigan & Wong Berkhin, 1979).

Este trabalho propõe uma abordagem híbrida para recomendação de conteúdo. Inicialmente será invocada a abordagem com Filtragem Baseada em Conteúdo e caso esta não tenha itens a serem recomendados, a abordagem baseada em Filtragem Colaborativa fornecerá o serviço. Com isto espera-se minimizar as deficiências de cada uma das abordagens em separado, como: (i) o problema do primeiro usuário e (ii) o problema de novos conteúdos. Em ambas as abordagens, como parte do processo de recomendação, propomos o uso da técnica de agrupamento, como discutido no Capítulo 3, na Seção 3.2.

4.3.3 *Semantic Modeling*

O módulo *Semantic Modeling* do componente RKTV é responsável por representar em uma ontologia o conteúdo multimídia minerado. Portanto, ontologias são utilizadas para a estruturação formal do conhecimento, tal como o raciocínio automático sobre estes. Neste sentido, propomos uma ontologia que represente o conhecimento resultante da clusterização dos dados e permita o raciocínio automático para a geração da recomendação de conteúdo.

Neste sentido, propomos o mapeamento dos conceitos da tarefa de clusterização para ontologias:

- Os elementos de um mesmo cluster terão a mesma raiz na estrutura da ontologia;
- O identificador do cluster será mapeado para o conceito de classe na ontologia;
- Os elementos de cada cluster serão mapeados para instâncias da classe que representa o cluster na ontologia.
- O conceito de usuário será mapeado para uma classe da ontologia, sendo esta integrante de um cluster;
- Cada classe de usuário terá a propriedade de assistir um programa;
- Os conceitos relativos a programação como: (i) Nome do programa, (ii) Horário, (iii) Horário que o usuário permaneceu assistindo a programação e (iv) Gênero; Serão mapeados para classes e terão relacionamentos com o STB do usuário.

Para descrever o processo de mapeamento vamos usar o seguinte formalismo:

- C_x é uma estrutura que representa um Cluster de identificador x;
- O é o conjunto de Objetos da base de dados;
- O_x é o conjunto de Objetos, tal que $O_x \subset O$, pertencentes ao cluster identificado por x, tal que $O_x \in C_x$;
- $O_{x,y}$ é um objeto $O \in C_x$ e tem como identificador y;

- Cl_x é uma Classe com identificador x da ontologia;
- I é uma Instância;
- I_x é o conjunto de Instâncias da Classe x;
- $\rho(Cl_x, O_x)$ função de mapeamento de um conjunto de objetos (O_x) para instâncias I_x de uma Classe Cl_x, $O_x \rightarrow I_x \in Cl_x$

Desta forma, o mapeamento proposto segue o pseudocódigo da Figura 24, tal que k é o número de clusters e n é o número de elementos de cada cluster.

Figura 24 – Mapeamento entre clusters e ontologias

```
for x=1 to x=k
{
 for y=1 to y=n
 {
      ρ(Cl_x, O_x,y)
 }
}
```

Neste sentido, na ontologia proposta, a OntoRKTV, são incorporados os conceitos definidos na coreKTV (Araújo, 2011), uma ontologia de programas de TV em uso na plataforma KTV, que estrutura os conceitos relativos a programação.

Adicionalmente, neste trabalho foi especificada uma ontologia específica para mineração de dados, a OntoClusteringKTV, com foco na tarefa de clusterização, dado que em outro trabalho do projeto foi especificada uma ontologia para tarefas de associação (Patrício Júnior, 2012). Nas subseções a seguir, serão detalhadas as ontologias especificadas neste trabalho de mestrado.

4.3.3.1 Ontologia OntoRKTV

Alguns trabalhos (Gottgtroy et *al*., 2004; Nigro et *al*., 2008; Diamantini et *al*., 2009; Kim et *al*.; 2011) estão sendo realizados unindo as áreas de mineração de dados e de representação do conhecimento por meio de ontologias. Estes estão organizados em duas abordagens (Patrício Júnior, 2012):

1. Uso de ontologias para a mineração de dados: Abordagem onde ontologias inserem conhecimento ao processo de mineração de dados;
2. Da mineração de dados para as ontologias: Abordagem onde o conhecimento minerado é representado por meio de ontologias.

Neste trabalho é utilizada a abordagem da mineração de dados para as ontologias, onde o conhecimento proveniente do processo da tarefa de clusterização, é representado por meio de uma ontologia, chamada OntoRKTV (Onto – Ontologia e RKTV – *Recommender Knowledge TV*), possibilitando, assim o raciocínio automático sobre os dados e o compartilhamento de dados.

No contexto do projeto KTV ontologias estão sendo analisadas para diversos fins no âmbito de dados multimídia. No que concerne à mineração de dados, ontologias são utilizadas para estruturar o conhecimento proveniente da tarefa de associação (Patrício Júnior, 2012). Neste trabalho, o escopo é expandido para a tarefa de clusterização.

A OntoRKTV tem o objetivo de representar o resultado da mineração de dados, especificamente da tarefa de clusterização, realizada sobre dados multimídia no ambiente da TVD interativa conectada. Basicamente este trabalho analisa e utiliza a OntoRKTV no processo de recomendação de conteúdo.

De forma imediata, o uso da OntoRKTV possibilita à plataforma KTV e a demais ambientes em que for utilizada:

- Automatização do raciocínio sobre os dados multimídia, tal como a possibilidade de inferência sobre os dados minerados;
- Análise das informações mineradas, identificando similaridades de conceitos;
- Reusabilidade do conhecimento descoberto por meio da mineração de dados, aplicando-os em diversos domínios, como o de *marketing* personalizado, grade de programação personalizada e recomendação de conteúdo;
- Possibilidade de extensão do conhecimento descoberto por meio da mineração de dados, agregando semântica aos dados, originalmente, apenas com sintaxe definida;

- Compartilhamento de dados, assim como da estrutura utilizada para descrição do conhecimento descoberto por meio da tarefa de clusterização.

Para a construção da OntoRKTV se fez necessário o uso de uma metodologia como guia (Noy & McGuinness, 2001; Fernández-Lópes & Gómez-Pérez, 2002). Esta será discutida na próxima subseção.

Metodologia de Criação da OntoRKTV

Dentre a gama de metodologias propostas para a construção de ontologias (Fernández-Lópes & Gómez-Pérez, 2002), neste trabalho utilizou-se a metodologia 101 especificada por Noy e McGuiness (Noy & McGuinness, 2001) por ser uma metodologia já estabelecida e ter sua implementação e documentação baseada na ferramenta Protegé (PROTEGÉ, 2012), utilizada para a construção das ontologias do projeto KTV e consequentemente da OntoRKTV.

A Metodologia 101 prega a construção de ontologias num processo iterativo de 7 (sete) passos: (i) Determinar o domínio e escopo da ontologia, (ii) Considerar o reuso, (iii) Listar termos, (iv) Definir classes, (v) Definir propriedades, (vi) Definir restrições e (vii) Criar instâncias.

Seguindo as etapas definidas pela metodologia 101, inicialmente foi definido o escopo da ontologia, que no caso da OntoRKTV é a representação do conhecimento minerado por meio de técnicas de clusterização, sendo o domínio os metadados multimídia da TVDi.

Em seguida, para fazer uso de ontologias já existentes, reusando-as, foram analisadas ontologias sobre o domínio de mineração de dados. Neste sentido foram encontradas as ontologias: DMOP – *Data Mining Optimization* (Hilario et *al*., 2011) e KDDOnto (Diamantini et *al*., 2009) que contêm diversos conceitos de mineração de dados, e embasam decisões de escolha sobre as tarefas de mineração, mas não apresentam conceitos específicos acerca de tarefas de clusterização e sua conexão com dados multimídia, ou seja, formalização da tarefa de clusterização de dados multimídia.

Ainda no contexto de reuso de ontologias foram avaliadas ontologias com conceitos de dados multimídia, que descrevem conceitos e relacionamentos entre

programas. Neste sentido, analisamos a ontologia coreKTV (Araújo, 2011), em uso na plataforma KTV, e a ontologia *BBC Programmes Ontology* (BBC, 2012), utilizada em alguns trabalhos pela comunidade da Web Semântica (Dogdu & Battal, 2010).

Outros trabalhos inseridos no projeto KTV, estão investigando um possível *matching* (Doan et *al.*, 2003). entre a ontologia coreKTV e a *BBC Programmes Ontology*, analisando as vantagens e desvantagens de cada uma. Para este trabalho, elegemos a primeira para ser adaptada e integrada aos conceitos de clusterização, por já ser utilizada na plataforma KTV por diversos projetos que estão sendo desenvolvidos em paralelo com este trabalho.

Seguindo as etapas definidas na metodologia 101, foi realizado o levantamento dos termos mais relevantes ao domínio da OntoRKTV. Neste caso, ocorre uma integração entre os conceitos de conteúdos multimídia e os de técnicas de clusterização. A seguir, o detalhamento dos termos identificados:

- Cluster: Agrupamento de STB com uso semelhante por turno;
- Membros: Elementos que pertencem a determinado cluster;
- STB: Um tipo de membro que pode ser pertencente a um cluster;
- Histórico: Os membros dos clusters possuem históricos de uso;
- Vizinhos: Membros de um mesmo cluster são vizinhos, possuem conteúdos em comum;
- Turno: Os membros de um cluster são definidos pelo padrão de uso por turno, podendo ser manhã, tarde, noite ou madrugada;
- Identificador do cluster: Identifica, nomeando, cada um dos clusters criados;
- Identificador dos membros: Identifica, nomeando, cada membro específico de um cluster.

Esses termos são o princípio para modelagem de classes, propriedades e instâncias da OntoRKTV. Esta busca modela o contexto de pós-processamento de KDD no que diz respeito aos agrupamentos gerados.

Os termos específicos do conteúdo multimídia, que compõem o histórico do STB, foram definidos em (Araújo, 2011), onde foi específicada toda a coreKTV, utilizada na plataforma KTV.

Desta forma as principais classes e grupo de classes modeladas na OntoRKTV, são (Figura 25): (i) *Cluster*, (ii) *History*, (iii) *PeriodOfDay*: *Morning, Night, Afternoon* e *Night* e (iv) *STB*.

Para realizar a conexão entre os conceitos da OntoRKTV, tal como entre a OntoRKTV e a coreKTV, foram criadas propriedades de objetos (Figura 25), retratando assim as ligações entre as entidades do mundo real, são elas:

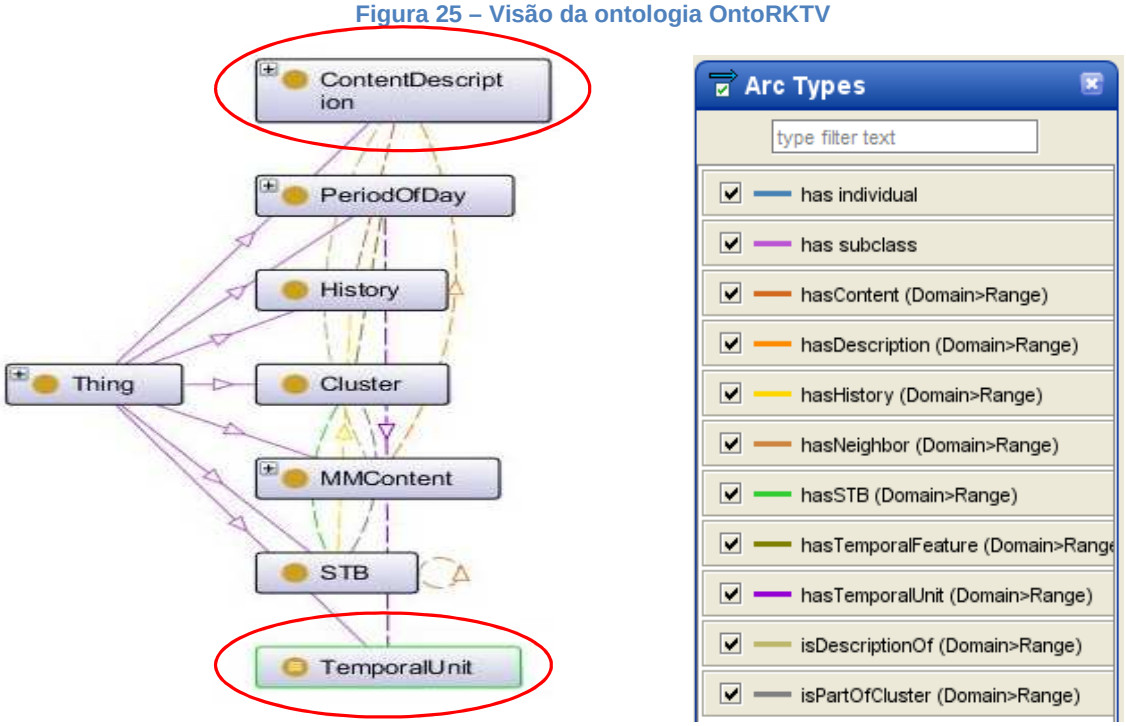
- *hasSTB*: Possui a classe *Cluster* como domínio e *STB* como contradomínio;
- *isPartOfCluster*: Possui a classe *STB* como domínio e *Cluster* como contradomínio. É a propriedade inversa de *hasSTB*;
- *hasHistory*: Possui a classe *STB* como domínio e *History* como contradomínio;
- *hasContent*: Possui a classe *History* como domínio e *ContentDescription*, da coreKTV, como contradomínio;
- *hasTemporalFeature*: Possui a classe *Cluster* como domínio e PeriodOfDay como contradomínio;
- *hasTemporalUnit*: Possui a classe *PeriodOfDay* como domínio e *TemporalUnit, da* coreKTV, como contradomínio;
- *hasNeighbor*: Possui a classe *STB* como domínio e *STB* como contradomínio;

As classes *ContentDescription* e *TemporalUnit* (Araújo, 2011), definidas na coreKTV, são o link para os conceitos de clusterização para a construção da OntoRKTV. *ContentDescription*, ou descrição de conteúdo, é a classe que representa um ponto chave agregador de diversas informações descritivas acerca do conteúdo transmitido. A classe *TemporalUnit* detalha questões de tempo sobre a programação. Na Figura 25 é ilustrada parte da OntoRKTV, em destaque as classes

definidas na ontologia coreKTV, as demais classes descritas na coreKTV foram suprimidas para facilitar a visualização dos conceitos adicionados.

Adicionalmente, foram definidas algumas propriedades de dados, conectando um indivíduo a um valor: (i) stbId, propriedade da classe *STB* que a identifica, (ii) clusterId, propriedade da classe *Cluster* que a identifica.

Figura 25 – Visão da ontologia OntoRKTV



Instâncias da OntoRKTV serão explanadas no Capítulo 5, onde serão realizados experimentos que validam o processo de recomendação proposto neste trabalho de mestrado.

4.3.3.2 Ontologia OntoClusteringKTV

Com o objetivo de automatizar o processo de mineração de dados por técnicas de clusterização, foi especificada e construída a ontologia OntoClusteringKTV, seguindo a mesma metodologia utilizada na construção da ontologia OntoRKTV, a metodologia 101 (Noy & McGuinness, 2001).

O escopo da OntoClusteringKTV é representar os conceitos da tarefa de clusterização. Foram pesquisadas ontologias com propósitos no mesmo domínio, com o objetivo de reuso. Adicionalmente às ontologias DMOP e KDDOnto, explanadas na Seção anterior, foi avaliada a ontologia OntoMKTV (Ontologia *Mining Knowledge TV*) (Patrício Júnior, 2012), utilizada na plataforma KTV, que trata de representar o

conhecimento minerado por meio de regras de associação. Nenhuma das ontologias avaliadas apresentam conceitos específicos acerca de regras de associação.

Nesse sentido foi realizado o levantamento dos termos mais relevantes ao domínio de clusterização de dados. Neste caso o OntoClusteringKTV tem como principais conceitos:

- Algoritmo de mineração de dados (EM, *K-Means*, ...);
- Tarefa de mineração (Clusterização);
- Classificação de algoritmos (Hierárquicos ou particionais);
- Cluster;
- Centroide;
- Instâncias do cluster;
- Métrica de similaridade.

Estes termos foram utilizados como princípio para modelagem das classes, propriedades e instâncias da OntoClusterigKTV, pois ela busca representar os conceitos da tarefa de clusterização. Desta forma as principais classes e grupo de classes modeladas, são (Figura 26):

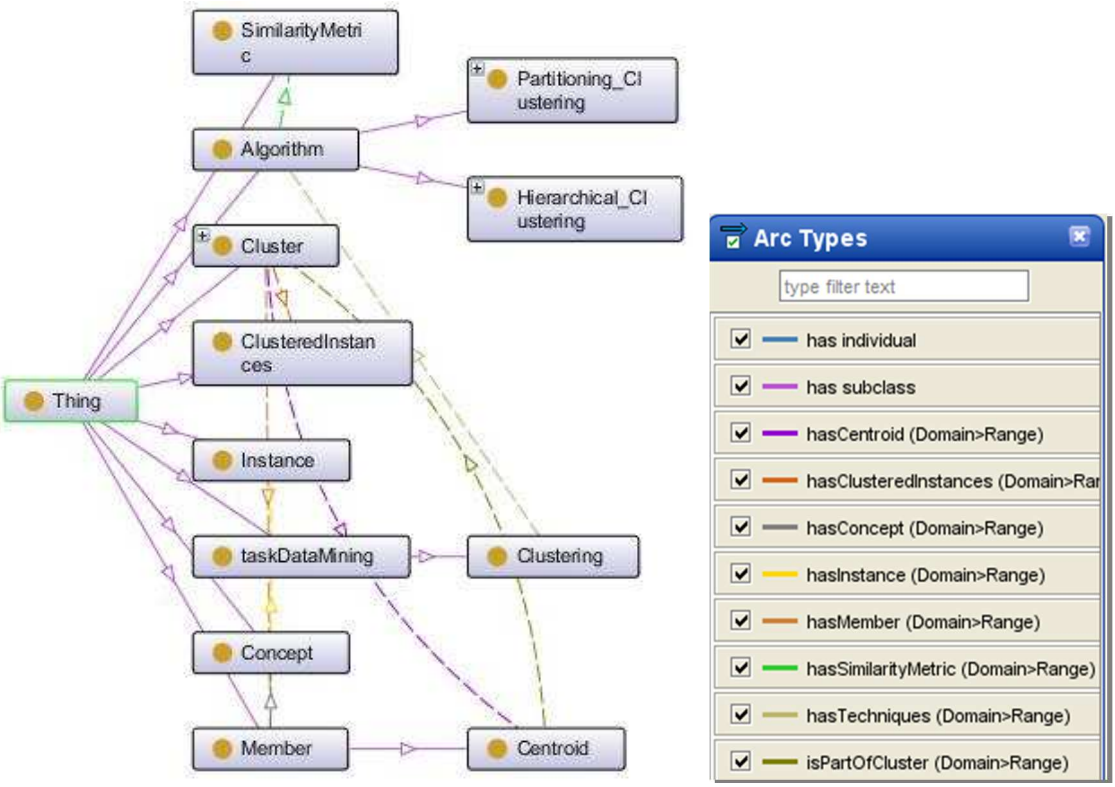
- *Algorithm*: Um algoritmo em geral, é uma sequência bem definida de passos que especifica a forma de resolver um problema ou executar uma tarefa;
- *Hierarchical_Clustering*: Classe que representa os algoritmos de clusterização classificados como hierárquicos;
- *Partitioning_Clustering*: Classe que representa os algoritmos de clusterização classificados como de partição;
- *TaskDataMining*: Classe que especifica a categoria de padrões que deverão ser encontrados;
- *Clustering*: Uma das tarefas de mineração de dados;
- *Cluster*: Representa um agrupamento com características semelhantes;
- *Member*: Representa os membros pertencentes a um cluster;
- *Concept*: Representa um atributo minerado através da ferramenta de mineração. Este atributo é genérico o suficiente para ser uma classe ou conceito de outra ontologia;

- *Instance*: Classe que se refere ao valor de um atributo ou a instância de um conceito, pode se remeter também a uma tupla em um banco de dados;
- *Centroid*: Um membro específico que representa o cluster como um todo;
- *Similarity_Metric*: Métrica utilizada pelo algoritmo para calcular a similaridade entre os membros de um cluster;
- *ClusteredInstances*: Indica a porcentagem de membros totais da base que pertence a cada dos clusters gerados.

Como destacado na Figura 26, algumas propriedade foram criadas para realizar a conexão entre os conceitos da OntoClusteringKTV, são elas:

- *hasCentroid*: Possui a classe *Cluster* como domínio e *Centroid* como contradomínio. A propriedade inversa de hasCentroid é isPartOfCluster;
- *isPartOfCluster*: Propriedade inversa de hasCentroid. Possui a classe *Centroid* como domínio e *Cluster* como contradomínio. Sendo uma especialização da propriedade *isPartOf*;
- *hasClusteredInstances*: Possui a classe *Cluster* como domínio e *ClusteredInstances* como contradomínio;
- *hasMember*: Possui a classe *Cluster* como domínio e *Member* como contradomínio;
- *hasConcept*: Possui a classe *Member* como domínio e *Concept* como contradomínio;
- *hasInstance*: Possui a classe *Member* como domínio e *Instance* como contradomínio;
- *hasSimilarityMetric*: Possui a classe *Algorithm* como domínio e *Similarity_Metric* como contradomínio;
- *hasTechiniques: Possui a classe Clustering como domínio e Algorithm como contradomínio.*

Figura 26 - Visualização OntoClusteringKTV

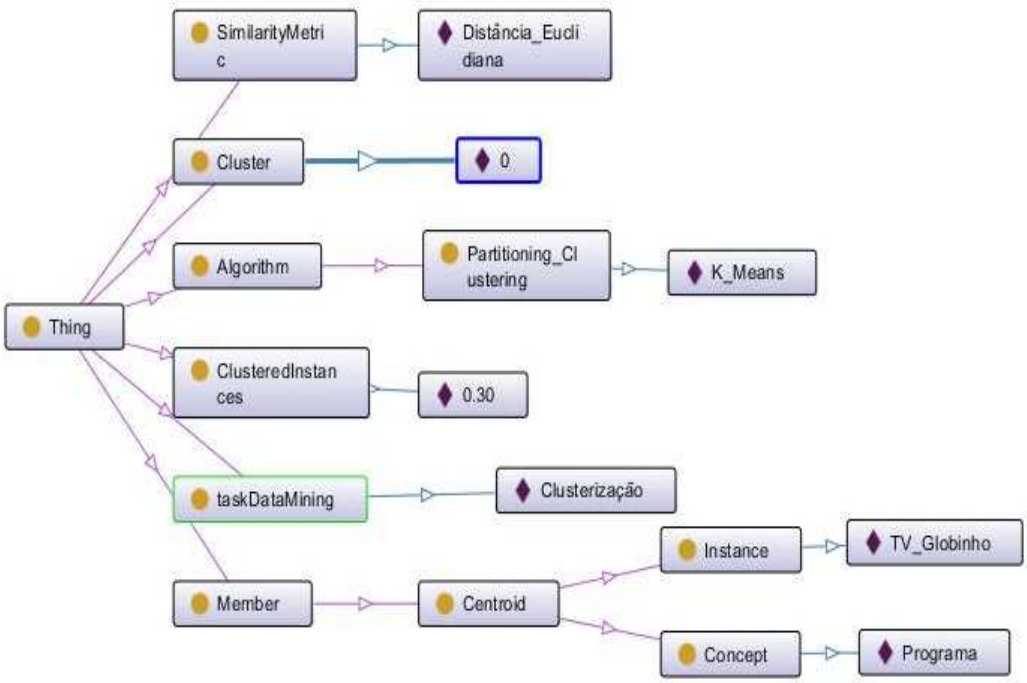


Um exemplo de instância que poderá ser representada pela OntoClusteringKTV é a seguinte (Figura 27):

- Tarefa de Mineração de Dados – Clusterização;
- Algoritmo de Mineração – *K-means*;
- Métrica de Similaridade – Distância Euclidiana;
- Cluster – 0;
- Centroide – Conceitos: Canal, Dia da Semana, Período do Dia, Programa, Gênero. Respectivamente aos conceitos tem-se as seguintes instâncias: GLO, Quarta-feira, Manhã, TV Globinho, Variedades;
- Instâncias do Cluster: 30%;
- Membros - Diversos dados que correspondem a 30% do total de membros da base de dados;
- Tipo de Algoritmo - Algoritmo baseado em Partição.

Para facilitar o entendimento da Figura 27, foram omitidos os relacionamentos entre as classes, resumido os atributos do centroide e destacado por um losango as instâncias.

Figura 27 - Representação de uma instância da OntoClusteringKTV



4.3.4 *Automated Reasoning*

A partir da ontologia gerada por meio do mapeamento proposto, poderá ser fornecida a recomendação de conteúdo de duas formas distintas: (i) Aplicando a tarefa de clusterização sobre os dados das emissoras (ii) Aplicando a tarefa de clusterização sobre os históricos dos STB. Em ambas, ontologias serão geradas e possibilitarão o raciocínio automático para a recomendação de conteúdo.

Basicamente, a proposta de recomendação almejada segue a expressão 1. A recomendação para um STB de identificador n, $R(stb_n)$, será uma escolha aleatória do resultado da subtração entre dois conjuntos, o conjunto de Objetos de todo o cluster k, ao qual o STB de identificador n pertence, e os Objetos visualizados no STB de identificador n $I(C_k)$ (Equação 1). Desta forma, conteúdos não conhecidos por um usuário e que são de seu interesse são recomendados.

$$R(stb_n) = I(C_k) - I(stb_n) \qquad (1)$$

Com o objetivo de aperfeiçoar a recomendação $R(stb_n)$, definida na Equação 1, foram definidas regras para o raciocínio automático sobre a OntoRKTV.

4.3.4.1 Regras Semânticas

Em princípio, propomos algumas regras e seus usos, utilizando-se dos relacionamentos semânticos disponíveis na OntoRKTV: (i) Equivalência, (ii) Especialização, (iii) Generalização, (iv) Instâncias de mesma classe, (v) Cluster e (vi) Vizinhança. Para a implementação destas regras foi utilizado o raciocinador Jena (Jena, 2013).

O objetivo principal do RKTV em disponibilizar algumas regras sobre a OntoRKTV é demonstrar como a ontologia pode ser utilizada para recomendação de conteúdo. Desta forma, outras regras de recomendação de conteúdo podem ser geradas a partir da OntoRKTV.

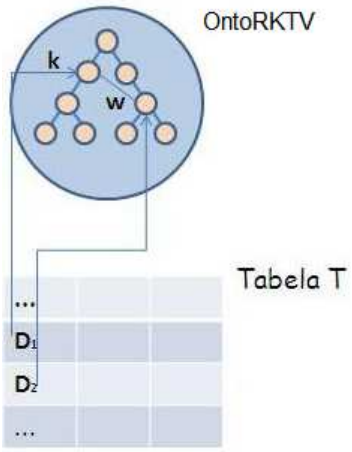
É importante destacar que neste trabalho de mestrado não foi avaliado quais regras trazem os melhores resultados de recomendação de conteúdo, apenas foram realizados experimentos que demonstram a aplicabilidade destas regras no contexto da TVDi (Capítulo 5).

I. Equivalência

Dois dados são considerados equivalentes se eles representam um mesmo conceito no mundo real.

Definição 1 - **Regra *isEquivalentTo***: um dado *D1* é equivalente a um dado *D2* se (i) *D1* é idêntico a um conceito k na ontologia OntoRKTV; (ii) *D2* é idêntico a um conceito w na mesma ontologia; e (iii) k e w estão ligados por um relacionamento *equivalentClass* na OntoRKTV. A Figura 28 ilustra a regra de equivalência.

Figura 28 – Regra de equivalência



Neste sentido, pode-se realizar raciocínio, por exemplo, sobre gêneros de programas, recomendando, portanto, programas com gênero equivalente ao que o usuário costuma visualizar. No contexto da OntoRKTV, que estende a CoreKTV, tem-se que o gênero ‘gospel’ é equivalente ao gênero ‘religioso’.

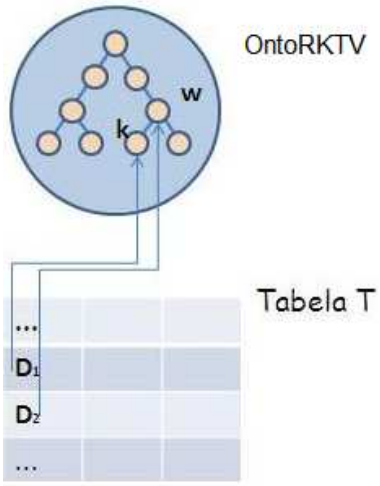
II. Especialização

Um dado é considerado uma especialização de outro dado se o primeiro for menos geral que o segundo.

Definição 2 - **Regra *isSubConceptOf***: um dado D_1 é uma especialização de um dado D_2 se (i) D_1 é idêntico a um conceito k na ontologia OntoRKTV; (ii) D_2 é idêntico a um conceito w na mesma ontologia; e (iii) k e w estão ligados por um relacionamento *subClassOf* na ontologia OntoRKTV. A Figura 29 ilustra a regra de especialização.

De acordo com esta regra de especialização, pode-se realizar raciocínio, por exemplo, sobre gêneros de programas, recomendando, portanto, programas com gênero em um subdomínio que o usuário costuma visualizar. No contexto da OntoRKTV, tem-se que ‘humor’ é uma especialização de ‘entretenimento’.

Figura 29 - Regra de especialização

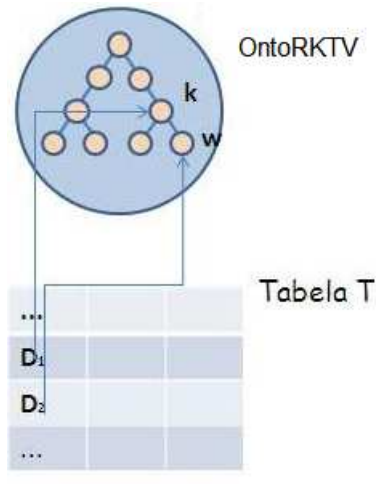


III. Generalização

Um dado é considerado uma generalização de outro dado se o primeiro for mais geral que o segundo.

Definição 3 - **Regra *isSuperConceptOf***: um dado D_1 é uma generalização de um dado D_2 se: (i) D_1 é idêntico a um conceito k na ontologia OntoRKTV; (ii) D_2 é idêntico a um conceito w na mesma ontologia; e (iii) k e w estão ligados por um relacionamento *superClassOf*. A Figura 30 mostra a regra de generalização.

Figura 30 - Regra de generalização



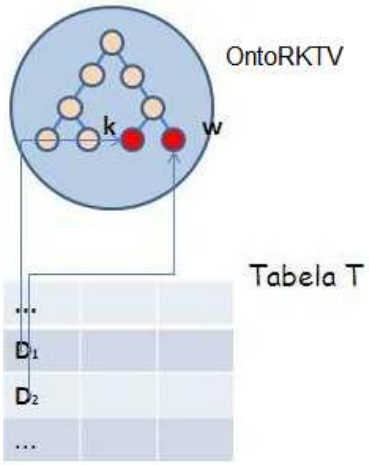
Por ser considerada uma regra oposta a regra de especialização, o raciocínio permitido é inverso ao anterior. Desta forma pode-se recomendar sobre gêneros de programas. No contexto da OntoRKTV, tem-se que 'entretenimento' é uma generalização de 'Reality'.

IV. Instância de mesma classe

Dois dados são considerados elementos de um mesmo conjunto se forem instâncias de uma mesma classe na ontologia de domínio.

Definição 4 - **Regra *isInstanceCloseTo***: Dois dados D_1 e D_2 são instâncias próximas se: (i) D_1 é idêntico a uma instância k na ontologia OntoRKTV e D_2 é idêntico a uma instância w na mesma ontologia; e (ii) k e w são instâncias do mesmo conceito. Na Figura 31, o esquema da regra de instâncias de mesma classe.

Figura 31 – Regra instâncias de mesma classe



V. Cluster

Usuários com comportamentos semelhantes visualizam programas de um mesmo gênero, com características similares. Esta característica permite definir que usuários similares pertencem ao mesmo cluster e consumem programas com gêneros que se aproximam. Desta forma, as recomendações podem ser executadas dentro de clusters em separado.

Definição 5 - **Regra *isPartOfCluster***: dois usuários U_1 e U_2 pertencem a um mesmo cluster se: (i) U_1 visualiza conteúdo semelhante ao conteúdo visualizado por U_2.

VI. Vizinhança

Um usuário possui vizinhos com um mesmo padrão de uso do STB, com costumes semelhantes. Usuários pertencentes a um mesmo cluster, provavelmente são vizinhos. O raciocínio automático pode explorar as vizinhanças de acordo com a

importância. Vizinhos mais interessantes são aqueles que possuem mais interseções de conteúdo, desta forma, recomendações entre eles pode obter resultados desejáveis.

Definição 6 - **Regra *neighbor***: U_1 é vizinho com forte interseção U_2 se: (i) O vizinho com maior número de conteúdos similares à U_2 é U_1; e (ii) U_1 pertence ao mesmo cluster que U_2.

4.3.4.1.1 Uso das Regras Semânticas

Regras semânticas são aplicadas na comparação semântica de dois conteúdos. A cada regra semântica foi associado um peso de acordo com sua importância no processo de recomendação, semelhante ao que foi realizado por Fernandez et al. (2006), no sistema AVATAR (Fernandez et al., 2006).

As regras de cluster e vizinhança são aplicadas apenas na abordagem com FC, no contexto deste trabalho não foram definidas precedências entre elas, dado que estas são resultantes qualitativamente do processo de clusterização. Seus usos são destacados no Capítulo 5, Seção 5.3.2.

As regras mais relevantes recebem um peso maior. Neste trabalho, usamos os seguintes pesos: 1.0 Instâncias da mesma classe, 0.8 para Equivalência e 0.7 para especialização e generalização, utilizadas para a abordagem com FBC e para a FC.

É importante destacar que os pesos das regras semânticas devem ser ajustados e avaliados. Neste trabalho de mestrado não foi possível realizar uma avaliação aprofundada sobre os pesos que otimizam a recomendação. Isto ocorreu por não termos o sistema operacional com usuários reais que avaliassem as recomendações sugeridas.

Basicamente como mais de um tipo de relacionamento semântico pode ser identificado entre dois conteúdos (O), o grau de similaridade semântica entre dois dados é dado pelo valor do maior peso dentre os diferentes tipos de relacionamentos semântico encontrados, como indicado na Equação 2.

$$S(O_1, O_2) = \max(weightRule_1, weightRule_2, ..., weightRule_n) \qquad (2)$$

As regras semânticas são utilizadas diferentes quanto a abordagem de recomendação utilizada. A seguir são destacados detalhes do uso das regras com (i) FBC e com (ii) FC.

I. FBC

No contexto deste trabalho de mestrado a FBC pode ocorrer em dois casos: (a) Com novos usuários, que não possuem histórico. (b) Usuários com histórico. A seguir será explanado o uso das regras semânticas em cada uma das vertentes.

a. Usuários sem histórico de uso

Para especificação do uso das regras semânticas neste cenário, foram seguidas as diretrizes do Capítulo 3, Seção 3.2.1.1. Neste caso, as regras semânticas serão aplicadas no cluster x de conteúdo que se encontra o conteúdo y que estava sendo visualizado no STB.

Para cada conteúdo pertencente a x será aplicada a Equação 2 com relação a y. Ao término, será eleito o conteúdo, ou grupos de conteúdo como maior similaridade semântica com y, como indicado na Equação 3, que utiliza a Equação 2.

$$R(STB_n) = \max(S(O_{(x,y)}, O_x - O_{(x,y)})) \qquad (3)$$

Na Equação 3, tem-se que $R(STB_n)$ é a recomendação para um STB de identificador n. $O_{(x,y)}$ é o conteúdo y pertencente ao cluster x. O_x são todos os conteúdos pertencentes ao cluster x. Neste sentido, para a execução da Equação 2 na Equação 3, exclui-se o próprio objeto em estudo ($S(O_{(x,y)}, O_x - O_{(x,y)})$).

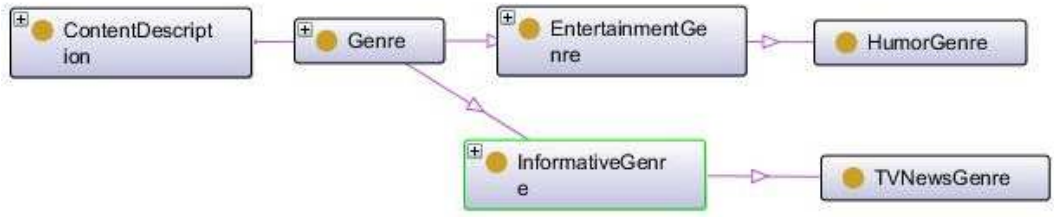
Para exemplificar, calculamos o grau de similaridade semântica entre o programa “TV Xuxa”, do gênero “Entretenimento” e os programas definidos na Tabela 3. A ontologia utilizada é OntoRKTV, extensão da CoreKTV (Figura 32). De acordo com esta ontologia, o tipo de relacionamento semântico encontrado entre “Entretenimento” e: (i) “Humorístico” é especialização, com peso 0.7. (ii) “Entretenimento” é instância de mesma classe com pesos 1.0. (iii) “Jornalístico”, que não possui regra definida neste trabalho, dado que só foi considerado um nível de ancestral, é 0.0.

Conforme mostrado na Equação 3, para determinar o grau de similaridade elege-se o que obteve maior valor de similaridade, neste caso, indicaria-se o programa “Caldeirão Huck”.

Tabela 3 – Exemplificação de conteúdos multimídia

Programa	Gênero
Chapolin	Humorístico
Caldeirão do Huck	Entretenimento
Jornal Nacional	Jornalístico

Figura 32 - Fragmento da OntoRKTV, extensão da CoreKTV



b. Usuários com histórico de uso

Para especificação do uso das regras semânticas neste cenário, foram seguidas as diretrizes do Capítulo 3, Seção 3.2.1.2. Neste caso, as regras semânticas serão aplicadas nos clusters que possuem históricos do usuário x, que requisitou o serviço.

A partir do conjunto de conteúdos que compõem os clusters que possuem históricos de x, seleciona-se: (i) os que possuem horários próximos de exibição e (ii) não faz parte do histórico do usuário. Sobre os dados resultantes são aplicadas as regras semânticas destacadas neste trabalho de mestrado. Para facilitar o entendimento, estes conjuntos resultantes foram nomeados como conjuntos alvo, conteúdos com possibilidade de recomendação.

Para cada conteúdo pertencente ao conjunto alvo, calcula-se a similaridade semântica deste com cada conteúdo do histórico do usuário x (STB), que está no mesmo cluster deste conjunto alvo (Equação 4). Posteriormente calcula-se a média

para obter-se a similaridade de um conteúdo alvo específico ($A_{(x,y)}$, elemento alvo y do cluster x) com o histórico do cluster x (H_x) do usuário.

$$S(H_x, A_{(x,y)}) = \frac{\sum_{i=0}^{i=n}(S(A_{(x,y)}, H_i)}{n} \qquad (4)$$

Posteriormente, de cada cluster são selecionados os conteúdos alvos que obtiveram maior índice de similaridade, como destacado na Equação 5.

$$R(STB_n) = \max(S(H_x, A_{(x,y)})) \qquad (5)$$

Este processo é repetido para cada um dos elementos do conjunto alvo e em cada um dos clusters selecionados. Ao término, tem-se de cada cluster que possui histórico do STB, os conteúdos mais similares ao seu uso e que estão próximos de serem exibidos.

Esta abordagem minimiza a possibilidade da recomendação para o STB ser sempre dos mesmos tipos de conteúdos, pertencentes a um mesmo cluster de conteúdos.

Para exemplificar, supondo que foram selecionados 2 clusters com conteúdos alvos a serem recomendados a um usuário x (Tabela 3). No primeiro cluster foram selecionados dois conteúdos alvos: “Chapolin” e “Caldeirão do Huck”. No segundo: “Jornal Nacional”. Em cada cluster foram executadas as regras semânticas. Como resultado, obteve-se que o programa “Chapolin” teve 0.4 de similaridade, representante do cluster 1, e “Jornal Nacional” teve 0.6 de similaridade. Neste sentido, seria recomendado o “Jornal Nacional”, de acordo com o histórico do STB.

II. FC

Para especificação do uso das regras semânticas neste cenário, foram seguidas as diretrizes do Capítulo 3, Seção 3.2.2. Neste caso, inicialmente são aplicadas as regras de vizinhança e cluster. Posteriormente, as demais regras são executadas, seguindo ideias semelhantes as da FBC.

Neste cenário as regras semânticas são aplicadas ao cluster que possuir o vizinho de maior similaridade de uso do STB com o usuário x, que requisitou o serviço.

As regras de especialização, generalização, instâncias de mesma classe e equivalência, serão aplicadas para verificar a similaridade semântica entre os conteúdos alvos, sugeridos pelo vizinho mais similar e a grade de programação atual. Este requisito se fez necessário, dado que muitos programas que poderiam estar no histórico dos usuários poderiam não mais estar sendo ofertados na programação.

Para diminuir o número de comparações, são realizadas comparações apenas com os próximos programas da grade, não realizando, assim, uma recomendação em longo prazo, como para dias e semanas.

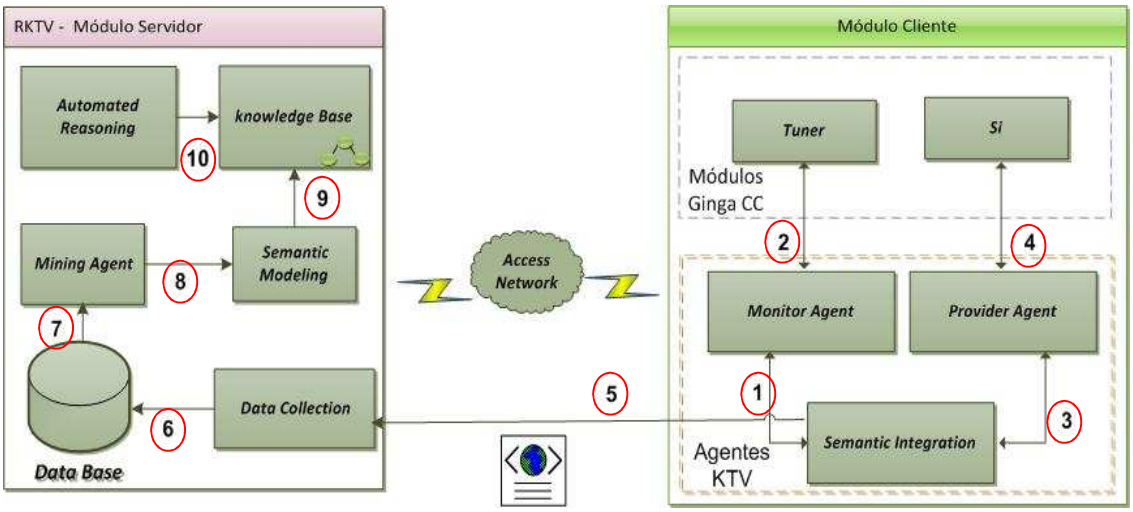
A similaridade é calculada da mesma forma que na FBC com histórico do usuário. Diferindo apenas que aqui são utilizados como conteúdos alvo, programas visualizados em um outro STB, adequando-os a programação atual da TV.

4.4 Fluxos de Dados do RKTV

Os fluxos de dados do RKTV estão destacados na Figura 33. Nesta, as interações entre módulos estão enumeradas para facilitar o detalhamento da estrutura.

Os metadados referentes às visualizações do usuário e dos conteúdos visualizados são estruturados e enviados pera o servidor por meio do módulo *Semantic Integration* do componente cliente. O *Semantic Integration* aciona o *Monitor Agent* (1), que detecta o canal em que o usuário está sintonizado. Em seguida o *Semantic Integration* envia para o *Provider Agent* as informações coletadas via *Monitor Agent* e solicita os metadados do conteúdo (3). Os metadados são estruturados e enviados periodicamente para o servidor por meio do *Semantic Integration* (5). Este envio é realizado via Web no formato XML (Figura 33).

Figura 33 – Fluxos de dados do RKTV



No servidor RKTV, os dados operacionais são recepcionados no módulo *Data Collection* (5), como destacado na Figura 33. Em seguida os dados são armazenados em um banco de dados relacional (Han & Kamber, 2006) (6), sendo submetido progressivamente ao processo de extração, transformação e carga (em inglês, *Extract Transformation and Load* – ETL) (Han & Kamber, 2006). Este foi detalhado, no contexto da plataforma do KTV, no trabalho de mestrado de um dos integrantes do grupo (Patrício Júnior, 2012).

Periodicamente os dados analíticos são submetidos ao processo de clusterização (7). Após a realização da mineração de dados o conhecimento descoberto será retornado atualizando as ontologias presentes na base de conhecimento de acordo com a modelagem especificada na camada de modelagem semântica (8). Os dados semânticos são armazenados em formas de ontologias no módulo *Knowledge Base* (9). Quando for solicitado o serviço de recomendação, o raciocínio automático é realizado sobre as ontologias (10).

4.5 Acesso aos Dados

Para acesso aos serviços da plataforma KTV, dentre eles o de recomendação de conteúdo definido neste trabalho de mestrado, é necessário a disponibilização de uma API (API KTV) baseada no conceito de *Web Service*.

No contexto do serviço de recomendação de conteúdo, a API KTV precisa ser estendida para incorporar operações específicas do processo definido nesta

dissertação. Desta forma, disponibilizam-se os conhecimentos descobertos no processo de mineração de dados.

A utilização de *Web Services* tem como vantagem a integração entre aplicações desenvolvidas por tecnologias diferentes. É importante destacar, que neste trabalho foi desenvolvida a aplicação local com o objetivo de validação dos conceitos definidos. Para utilizar a aplicação e necessário indicar o id do usuário que se quer recomendar conteúdo. Como resposta é disponibilizado um arquivo XML com o conteúdo a ser recomendado.

4.6 Considerações Finais

Neste Capítulo foi apresentado todo o processo de recomendação proposto por este trabalho de mestrado. Para isto foi realizado o detalhamento da arquitetura proposta, indicando todos os seus componentes, módulos e interações, tal como a interligação entre este trabalho e a plataforma KTV.

Neste sentido, foi realizado o detalhamento do componente cliente com os seus 3 (três) módulos: *Monitor Agent*, *Provider Agent* e *Semantic Integration*. Em seguida foi realizado o detalhamento do componente servidor e de seus módulos: *Data Collection*, *Mining Agent, Semantic Modeling*, *Automated Reasoning.*No módulo *Semantic Modeling* foram detalhadas as ontologias construídas durante este trabalho a OntoRKTV e a OntoClusteringKTV. No módulo *Automated Reasoning* foram definidas regras semânticas.

Por fim foi detalhado o fluxo de dados ao longo do processo de recomendação de conteúdo multimídia segunda a arquitetura RKTV. No próximo Capítulo são discutidos os experimentos que validaram esta proposta e discutidos os resultados alcançados.

Capítulo 5

Validação RKTV

Com o objetivo de validar a aplicabilidade do processo definido no RKTV da plataforma KTV, foram realizados experimentos para dar subsídios que, de forma prática, demonstrassem a aplicabilidade e viabilidade deste trabalho. Nesse sentido foi definida a base NetFlix (NetFlix, 2013), base de dados de uma empresa que aluga filmes online, por ser uma base pública e um subdomínio de TVDi. Nas próximas seções serão detalhados os experimentos realizados, a base de dados e as análises sobre os resultados obtidos, tal como uma instância para a arquitetura RKTV.

5.1 Instância da Arquitetura RKTV

A arquitetura RKTV é genérica (Figura 21), podendo ser instanciada por diversas técnicas e com o uso de diferentes tecnologias. Como forma de demonstrar a viabilidade da arquitetura, foi realizada uma implementação de 5 (cinco), dos 6 (seis) módulos previstos no componentes servidor. O módulo APIKTV, não foi desenvolvido durante este trabalho, pois não influencia diretamente nos experimentos realizados. Em seu lugar foi desenvolvido uma aplicação para o fim de validação, explicitada nas próximas Subseções e destacada no Apêndice B. Apesar do sistema está disponível, não foi desenvolvido como um serviço.

De acordo com a arquitetura proposta do RKTV (Capítulo 4), o módulo *Semantic Integration* é responsável por integrar os dados do componente cliente e submetê-los em formato XML para o servidor KTV. Este trabalho não tem como foco o desenvolvimento dos agentes integrados ao *middleware* Ginga. Estes estão sendo finalizados por outros integrantes do projeto KTV.

O protótipo MKTV (Patrício Júnior, 2012), desenvolvido da linguagem de programação Java (Java, 2013), integra algumas tecnologias utilizadas no RKTV: (i) XML. (ii) PostgreSQL (PostgreSQL, 2012). Basicamente o MKTV possui uma interface web, que simula o envio de matadados em XML pelo *Semantic Integration*, componente cliente, e o mapeamento para o banco de dados relacional realizado pelo módulo *Data Collection*, componente servidor.

O módulo *Mining Agent*, foi desenvolvido em Matlab (Matlab, 2013), desde o pré-processamento dos dados até a fase da tarefa de mineração de dados (k-means). Os principais trechos de códigos e parâmetros definidos estão destacados no Apêndice A. Na Seção 5.2, está descrito todo o funcionamento e implementação deste módulo.

As funcionalidades do módulo *Semantic Modeling* foram desenvolvidas por meio da ferramenta Protegé (Protegé, 2013), ontologias discutidas do Capítulo 4. Para raciocínio automático utilizou-se a API Jena (Jena, 2013).

Por fim, foi desenvolvido em Java o sistema baseado no RKTV que integra os módulos anteriores e implementa o processo de recomendação proposto neste trabalho de mestrado. No apêndice B, foram destacadas algumas classes do sistema, com foco nos parâmetros de entradas e saídas.

Apesar do foco deste trabalho não ser a especificação/implementação do componente Cliente da Figura 20. Este foi especificado e implementado em outro trabalho (Silva et *al*., 2013), tal qual as necessidades do RKTV, definido neste trabalho.

Nas próximas seções serão detalhadas as bases de dados utilizadas nos experimentos.

5.2 Experimentos com a Base do NetFlix

O Netflix (Netflix, 2013) aluga de forma online filmes e embasa o seu modelo de negócio sobre recomendações de filmes aos seus usuários. O sucesso das vendas ou assinaturas fica a cargo da eficiência das sugestões, pois caso o usuário não encontre o filme que deseja e não se sinta atraído a assistir um novo filme ele não utilizará mais o serviço.

A empresa incentiva seus clientes a expressarem opinião sobre o quanto eles gostaram dos filmes que assistiram e de 1998 à 2006 já havia recolhido cerca 1,9 bilhão de votos de mais de 11,7 milhões de usuários em mais de 85 mil títulos de filmes (Bennett & Lanning, 2007).

Em 2006 o Netflix propôs um desafio para a comunidade científica com o intuito de aperfeiçoar o seu sistema de recomendação. Neste sentido disponibilizou uma base de dados de classificação com mais 480 mil classificações de mais de 18 mil títulos de filmes.

A utilização da base de dados disponibilizada pelo Netflix se dá devido as informações trabalhadas pelo seu sistema, com avaliações e históricos de usuários. Adicionalmente, a base do Netflix faz parte de um subdomínio da TVDi, filmes. Estas características agregam valor a esta pesquisa, principalmente por ser uma base de dados real.

5.2.1 Análise dos Dados NetFlix

A base original de dados do Netflix disponibiliza as seguintes informações:

- *IDFilme*: Inteiro único atribuído arbitrariamente no intervalo [1 ... 17.770];
- *CustomerID*: Inteiro único atribuído arbitrariamente no intervalo [1 ... 2649429];
- *Rating*: Número de 'estrelas' atribuídos a um filme por um cliente. Número entre 1 e 5;
- *Title*: Título em inglês do filme no site do Netflix;
- *YearOfRelease*: Ano que um filme foi lançado no intervalo [1890 ... 2005]. Pode corresponder ao ano de lançamento do DVD correspondente;
- *Date*: Data de uma classificação na forma AAAA-MM-DD, na faixa de 1998/11/01-2006/12/31.

Os experimentos foram utilizados com a base original (com os atributos listados), e uma repetição do experimento com a base enriquecida. A base foi enriquecida com informações retiradas do DBPedia (DPPedia, 2013), tais como:

- Diretor: Nome do diretor responsável pelo filme;
- Ator: Atores que participarem do filma;
- Categoria: Categoria na qual o filme se enquadra.

5.2.1 Pré – Processamento dos Dados

Algumas convenções e pré-processamento foram realizadas sobre os dados, afim de aperfeiçoar a recomendação tal como adequar-se ao algoritmo k-means.

De acordo com a avaliação (*Rating*) as convenções foram: (i) Péssimo, para uma estrela. (ii) Ruim, para duas estrelas. (iii) Regular, para avaliação de valor três. (iv) Bom, para avaliação com valor 4. (v) Ótimo, para avaliação com valor cinco. Como nosso objetivo é recomendar colaborativa entre os usuários, optamos por selecionar de cada usuário apenas os filmes que foram avaliados como bons ou ótimos (rating 4 ou 5). Desta forma, os usuários recomendam entre si, filmes que julgaram bons.

Como queremos identificar grupos de usuários com perfis semelhantes optou-se por utilizar o nome do filme (experimento com o NetFlix original), e, adicionalmente, nome do diretor, nome ator e categoria, nos experimentos com a base enriquecida.

Para a realização do agrupamento de usuários, foi gerada uma matriz com uma coluna para o código do usuário e colunas para os filmes vistos por cada usuário. A Tabela 4 demonstra esta transformação. Cada linha da tabela representa um vetor de interesses de um usuário, sendo assim, dois usuários com interesses semelhantes possuem vetores próximos. Desta forma, na Tabela 4, tem-se que o usuário de identificação “69867”, assistiu ao filme “Congo”, uma vez, e aos filme “*Mississippi Burning"* e *"The Santa Clause"*, zero vezes.

Tabela 4 - Exemplo de vetores representativos dos interesses dos usuários

User_id	*"Congo"*	*"Mississippi Burning"*	*"The Santa Clause"*
69867	1	0	0
437	1	1	0

Para a base do NetFlix enriquecido, foi realizada a mesma transformação, mas neste caso, além das dimensões para os nomes dos filmes, teve-se as dimensões para os diretores, atores e categorias.

Analisando os dados, observou-se *Scarcity problem* (Seção 2.1.4.1), bastante conhecido em abordagens colaborativas, este problema se deu pela grande variedade de filmes a serem avaliados, gerando uma matriz esparsa de dados (muitas dimensões com poucas avalições).

Testes iniciais demonstraram que um grande número de dimensões prejudica os resultados encontrados pelo algoritmo de agrupamento. Isto é, os grupos identificados não são bem definidos, sendo estes de difícil classificação por causa da grande dispersão de seus dados em todas as dimensões.

Para comprovar esta teoria o algoritmo foi executado com todas as dimensões geradas, com k variando de 2 – 400 e o máximo de média de *Silhouette Coefficient* (Coeficiente de Silhueta) (Tan et *al*., 2006) foi 0,05, quando o indicado é superior a 0,5 (Kononenko & Kukar, 2007; Khalilian et *al*., 2010). Portanto, optamos por reduzir a variabilidade de filmes, selecionando apenas aqueles que foram visualizados por pelo menos 10% dos usuários do sistema.

Com o objetivo de realizar recomendações, a base foi fatiada por períodos, por meio do atributo “When”. Neste caso, executamos o algoritmo com os 6 primeiros meses de uso do sistema ("1999-12-08” - "2000-05-08").

Após o pré-processamento temporal, os dados foram sumarizados da seguinte forma:

- Avaliações: 23136
- Usuários: 3432 (vetores)
- Filmes: 187 (dimensões)

Após o pré-processamento com as demais restrições, os dados foram sumarizados da seguinte forma:

- Avaliações: 11889
- Usuários: 2916 (vetores)
- Filmes: 18 (dimensões)
- Atores:17 (dimensões)
- Diretores: 17 (dimensões)
- Categorias:17 (dimensões)

5.2.2 Clusterização Sobre os Dados do Netflix

Após o pré-processamento dos dados foram aplicadas as técnicas para a detecção do agrupamento de dados. O algoritmo foi realizado considerando o histórico de cada usuário.

Para o processamento do algoritmo *k-means*, o k variou de 2 – 300. Para diminuir a distorção causada pela obtenção de mínimos locais diferentes dependendo

da escolha dos k-centróides iniciais, foram realizados testes com 5 (cinco) sementes aleatórias diferentes para cada valor de k. Após o processo, foi selecionada a semente que retornou os melhores resultados para média do coeficiente de silhueta (Tan et *al*., 2006) e para o erro quadrático (Tan et *al*., 2006). A métrica de similaridade utilizada foi a distância euclidiana.

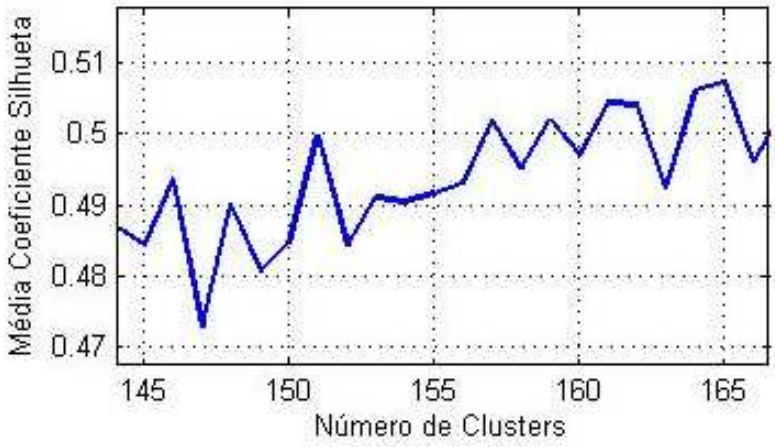
A média do coeficiente de silhueta foi avaliado segundo a tabela 5 (Kononenko & Kukar, 2007; Khalilian et *al*., 2010).

Tabela 5 - Análise da média do coeficiente de silhueta

0.70—1.0	Estrutura encontrada é a ideal
0.50-0.70	Estrutura encontrada é aceitável
0.25-0.50	Estrutura encontrada é fraca e pode ser artificial
<0.25	Nenhuma estrutura substancial foi encontrada

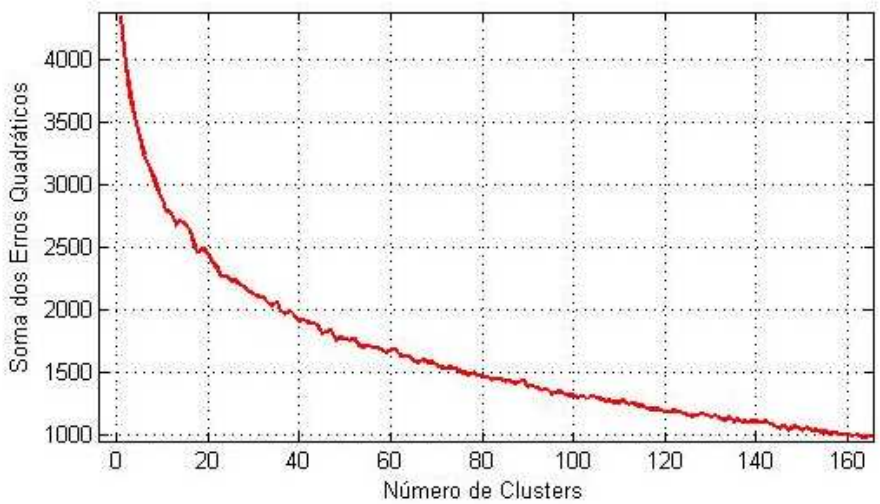
A partir de k = 157, as alterações na média do coeficiente de silhueta foram pouco significativas, com crescimento lento, como demostrado no gráfico da Figura 34. Neste sentido, os experimentos foram realizados com k=157.

Figura 34 - Gráfico média coeficiente silhueta



Além do coeficiente de silhueta, foi avaliada a medida de erro quadrático, que tende a zero, quando k tende a infinito (Tan et *al*., 2006). É importante destacar que a decisão do número de grupos é primordial. Não se deve quebrar um cluster em dois distintos, nem tão pouco unir clusters que essencialmente são distintos (Berkhin, 2002). Desta forma, avalia-se o erro quadrático onde ocorre um pequeno "joelho" e uma diminuição na velocidade que o erro decrementa (Tan et *al*., 2006), como destacado no gráfico da Figura 35.

Figura 35 - Gráfico Soma dos Erros Quadráticos



É importante destacar, que neste trabalho foram avaliadas duas medidas para seleção do k ideal, no entanto, estas são independentes, convergindo para um mesmo resultado aproximado.

Na execução do experimento com a base de dados enriquecida, obteve-se resultados semelhantes, não houveram melhoras significativas com relação ao coeficiente de silhueta, tão pouco uma redução na soma dos erros quadráticos. Isto se deu devido a características dos dados enriquecidos, poucos filmes possuem mesmo ator, diretor ou categoria. O enriquecimento realizado por meio do DBPedia, foi bastante específico, o que não influenciou positivamente no agrupamento dos dados.

5.2.3 Uso das Regras Semânticas sobre os Dados do Netflix

Após o processo de clusterização (agrupamento sintático), ao processo de recomendação do conteúdo foi integrado o uso das regras semânticas definidas e formalizadas na Seção 4.3.4.1. Estas regras foram implementadas e avaliadas neste trabalho de mestrado.

É importante destacar que neste trabalho foram especificadas duas ontologias a OntoKTV (extensão da CoreKTV) e a OntoClusteringKTV (para estruturar o processo de clusterização). Na Figura 27, foi demostrado uma instância da OntoClusteringKTV. Na Figura 26, foi demostrado uma instância da OntoKTV.

No contexto da base de dados avaliada neste trabalho, as regras foram aplicadas sobre os atributos que representam a categoria dos filmes e diretores. Considerando os dados da Tabela 6, apenas as regras I, iV, V e VI (Seção 4.3.4.1). Não ocorre especialização / generalização considerando os gêneros definidos na CoreKTV.

Tabela 6 – Exemplo de Regras aplicadas durante os experimentos

Nome	Categoria	Diretor
Yojimbo	***Jidaigeki***	***Akira Kurosawa***
Seven Samurai	***Fictional samurai***	***Akira Kurosawa***
Presumed Innocent	***Works by Scott Turow***	***Alan J. Pakula***
The Devil's Own	***Troubles***	***Alan J. Pakula***

No contexto deste trabalho as regras semânticas foram utilizadas com o objetivo de reduzir o grupo de filmes a serem recomendados, tal como aumentar a precisão do sistema, como descrito e avaliado na Seção 5.3.2.1

5.3.2 Recomendação Sobre os Dados do Netflix

No contexto da base de dados discutida, base pré-processada no Netflix, é possível realizar recomendação por filtragem colaborativa. Neste sentido pode-se gerar um cenário de uso que utiliza o conhecimento minerado, seguindo as especificações do Capítulo 3 na Seção 3.2.2 e as regras definidas no Capítulo 4 na Seção 4.3.4.1:

- Um usuário solicita a recomendação de conteúdos similares aos que ele costuma consumir;
- É realizada a detecção de qual cluster o usuário que solicitou a recomendação pertence. Supondo que a solicitação tenha sido realizada pelo usuário de identificação “746834”, pertencente ao cluster 2, Será detectado qual o usuário destes clusters possui mais similaridades com o usuário solicitante. Neste caso os usuários “1968605” e “2237957”. Em seguida são selecionados os filmes vistos por estes e não vistos por aquele

5.3.2.1 Análise dos resultados obtidos

Para analisar o k-means no serviço de recomendação de conteúdo, foi utilizada a métrica *Precision* (Precisão) (Montaner, 2003), conforme a Seção 2.1.5. No contexto

deste trabalho, foi realizada uma média da precisão da recomendação realizada a todos os usuários do NetFlix.

Neste sentido, para cada usuário, calculou-se o número de acertos das recomendações realizadas (por meio do histórico do usuário, que não estava no período avaliado na clusterização) sobre o número total de recomendações realizadas pelo RKTV. O processo descrito na subseção anterior foi replicado 10 vezes, com períodos diferentes. Em média, obteve-se uma precisão de 30,04% de acertos.

Adicionalmente, foram aplicadas as regras semânticas, como previsto no processo de recomendação de conteúdo proposto neste trabalho. Com uso das regras semânticas, obteve-se um refinamento do conjunto de filmes recomendados pelo RKTV, atingindo, em média, 35,12% na métrica de precisão. Este refinamento, se dá pela diminuição do conjunto de elementos a serem recomendados.

É importante destacar que o pequeno ganho de precisão da recomendação apenas com a clusterização e da recomendação seguida da avaliação semântica se deu pela características do enriquecimento semântico dos dados. Estes possuem poucas interseções de categorias, tal como especificidades que não estão modeladas na CoreKTV (Araújo, 2011). Neste sentido, poucas instâncias tomaram proveito dos conceitos modelados na ontologia.

5.4 Considerações Finais

Neste Capítulo foi demonstrada a aplicabilidade do processo de recomendação de conteúdo descrito deste trabalho de mestrado. Esta validação foi realizada aplicando-se o processo proposto em uma base de dados real, um portal de aluguéis de filmes sobre demanda, o Netflix.

Sobre a base de dados foram aplicadas técnicas de mineração de dados, *k-means*, que resultaram em grupos de dados similares provenientes das informações analisadas e realizados raciocínios segundo a ontologia OntoRKTV.

As informações provenientes da base de dados do Netflix, demonstraram o processo de recomendação colaborativa proposto neste trabalho de mestrado. Seguido do uso das regra semânticas. É importante destacar que o enriquecimento semântico realizado na base não aumentou significativamente a precisão da

recomendação, quando avaliada a mineração sintática dos dados. O uso de regras semânticas aperfeiçoou a recomendação, o que foi mostrado por meio da métrica *Precision*.

Os resultados obtidos demonstram a funcionalidade do RKTV no contexto do Projeto KTV. No próximo Capítulo serão discutidos os trabalhos relacionados que possuem interseções com este trabalho de mestrado.

Capítulo 6

Trabalhos Relacionados

No âmbito de sistemas de recomendação existe uma vasta quantidade de trabalhos no domínio da TVDi ou outras áreas de pesquisa. Estes trabalhos, além do domínio, diferem nas técnicas utilizadas e no propósito da recomendação. Neste Capítulo são apresentados trabalhos que apresentam interseções com a proposta apresentada, realizando um quadro comparativo ao final.

6.1 Fernandez et *al*. (2006)

Em Fernandez et *al*. (2006) é apresentado o sistema AVATAR, projetado para a TVDi conectada. Assim como neste trabalho, o AVATAR também utiliza coleta de dados implícita, com FH, utilizando os conceitos da Web Semântica.

No sistema AVATAR a abordagem com FBC objetiva a descoberta de relações semânticas entre o histórico do usuário com os conteúdos a serem recomendados, conteúdos alvo. Neste sentido, são utilizados dois tipos de relacionamentos: (i) Semelhança Semântica Hierárquica, que considera os relacionamentos de especialização de ontologias e (ii) Inferência de Semelhança Semântica, que realiza descoberta de relações implícitas entre objetos. A abordagem colaborativa do AVATAR utiliza *matching* de ontologias (Doan et *al.*, 2003).

Dentre as interseções estão a utilização de ontologias para representação de conhecimento no domínio da TV, coleta de informações implícitas sobre o usuário e a utilização de raciocínio automático no processo de filtragem baseada em conteúdo.

Como principais diferenças, está o fato de o AVATAR ser desenvolvido para o *middleware* MHP, enquanto que este trabalho propõe uma abordagem para o Ginga;

outra diferença é o fato de não serem utilizadas técnicas de mineração de dados no sistema AVATAR.

6.2 Hsu et *al*. (2007)

O sistema proposto por Hsu et *al*. (2007), o AIMED, realiza recomendação baseada nas informações coletadas e armazenadas no histórico do usuário e combina-as com informações demográficas e atividades de interesse. A recomendação é realizada por meio de técnicas de redes neurais artificiais que tentam prever as preferências dos usuários.

O AIMED é um sistema de recomendação híbrido, tanto na forma da coleta dos dados, quanto na abordagem de filtragem utilizada (FC E FBC). Este trabalho propõe recomendação híbrida, no entanto com coleta implícita de dados, o que diferencias os dois trabalhos.

Outro diferencial entre os trabalhos é o fato desta proposta ter foco no SBTVD, incorporando ao *Middleware* Ginga componentes que viabilizem o serviço de recomendação de conteúdo. Assim como, realizar recomendação pelo o uso do STB, sem a necessidade de preenchimento de cadastros.

6.3 Aroyo et *al.* (2007)

O trabalho apresenta o Sensee, um *framework* para personalização de TV, com experimentos baseados em TV na Web e com Set-Top-Boxes (Aroyo et *al.*, 2007).

O Sensee é baseado em semântica, utilizado ontologias para o gerenciamento de vocabulários, fazendo análise de itens que são sinônimos e possibilitando que o sistema entenda semanticamente os itens a serem recomendados. Adicionalmente, ontologias são utilizadas para estruturar o perfil do usuário, assim como para o uso de contexto (Bettini et *al.*, 2010), como: (i) tempo, (ii) localização e (iii) audiência.

Assim como neste trabalho, o *framework* proposto por Aroyo et *al*. (2007) considera a Web como meio para busca de metadados, no entanto, são desconsiderados os metadados provindos via *broadcast*. Estes metadados também são considerados neste trabalho de mestrado, como ilustrado da Figura 22. Outro diferencial é o fato de o *framework* Sensee não fazer uso de técnicas de mineração de dados; e não considerar o contexto do SBTVD.

6.4 Ávila (2010)

O trabalho apresenta o sistema *RecommenderTV* (Ávila, 2010) que possui muitas interseções com a abordagem apresentada nesta proposta, como: (i) Arquitetura baseada em agentes responsáveis pela coleta implícita de dados e (ii) Processo de recomendação para múltiplos usuários.

As principais diferenças concentram-se no processo de recomendação. Enquanto que o *RecommenderTV* realiza recomendação basicamente por técnicas de mineração de dados (regras de associação) e utiliza apenas FBC, não utilizando conceitos da Web Semântica. Este trabalho de mestrado utiliza FH e enriquecimento semântico que antecede a etapa de mineração por clusterização, tal como estrutura o conhecimento de forma a permitir o processamento automático sobre o conhecimento descoberto.

Outro ponto que difere as duas abordagens é a arquitetura. Neste trabalho de mestrado a arquitetura é baseada em um módulo cliente e outro servidor (onde ocorre majoritariamente o processamento para o serviço de recomendação), utilizando, assim, o canal de retorno da TVDi. Em contrapartida, no *RecommenderTV* todo o processamento é realizado no próprio STB da TVD.

6.5 Neto et *al*. (2010) Guia de Programação Eletrônico (GPE)

O sistema proposto por Neto et *al*. em Abordagem *Combinada para Recomendação Personalizada Utilizando o Guia de Programação Eletrônico* (Neto et *al.,* 2010) apresenta um Guia de Programação Eletrônico (GPE), utiliza Filtragem Baseada em Conteúdo por meio de mineração de dados por regras de associação.

Em Neto et *al.* é proposto um sistema com arquitetura centralizada apena no STB, com coleta híbrida, onde todo o histórico de uso do STB é estruturado em XML e é requerida a avaliação dos usuários quanto a qualidade da recomendação. Neste trabalho de mestrado é proposta uma arquitetura cliente-servidor, com coleta de dados implícita e uso de Filtragem Híbrida, características que diferenciam os dois trabalhos.

Outro diferencial é o fato de neste trabalho de mestrado considerar ontologias como o modelo para a estruturação do perfil dos usuários, tal como dos conteúdos a serem recomendados.

6.6 Kim et *al.* (2011)

O sistema de recomendação proposto por Kim et *al*. em *Recommendation System of IPTV TV Program Using Ontology and K-means Clustering* (Kim et *al.*, 2011) apresenta uma solução baseada em mineração de dados por clusterização.

Em Kim et *al.* é proposto um sistema com arquitetura cliente-servidor, onde no módulo cliente é estruturado todo o histórico de uso do STB em uma ontologia. No lado servidor esta base é enriquecida semanticamente. Estas características aproximam o trabalho de Kim et *al.* e a proposta deste trabalho, no entanto os trabalhos diferem no processo de recomendação, onde no trabalho de Kim et *al.* a clusterização ocorre entre as ontologias do histórico do STB por meio de uma função de similaridade. Enquanto que neste trabalho de mestrado a mineração de dados é realizada sobre os metadados da programação visualizada.

Outro diferencial é em relação ao tipo de filtragem utilizada. Em Kim et *al.* é realizada apena FC, enquanto que este trabalho de mestrado propõe uma abordagem híbrida.

6.7 Comparativo

De forma comparativa foram avaliados alguns pontos de cada um dos trabalhos analisados de acordo com algumas características (Tabela 9, 10):

- Tipo de Coleta;
- Tipo de Filtragem;
- Se o processo proposto de recomendação possui algum tipo de enriquecimento semântico;
- Se o processo de recomendação utiliza mineração de dados;
- O tipo de validação utilizado no processo;
- Se o processo utiliza canal de retorno;
- Se a proposta é incorporada a algum *middleware*.

Para facilitar a visualização das informações, a Tabela 9 foi replicada, gerando a Tabela 10 com a continuação das informações. A última coluna da Tabela 10 é referente às classificações deste trabalho de mestrado. Este é classificado como parcialmente integrado ao *middleware*, por necessitar dos metadados incorporados no *middleware* para fornecer o serviço de recomendação de conteúdo.

Tabela 7 – Quadro comparativo de trabalhos

	Fernandez et *al.*	Hsu et *al.*	Aroyo et *al.*
Coleta	Implícita	Híbrida	Implícita
Filtragem	Híbrida	Híbrida	FBC
Semântica	Sim	Não	Sim
Mineração de dados	Não	Não	Não
Validação	Não é explicitada no trabalho	Métricas	Métricas
Canal de Retorno	Sim	Não	Sim
Middleware	Sim	Sim	Não

Tabela 8 - Quadro comparativo de trabalhos (continuação)

	Ávila	Neto et *al.*	Kim et *al.*	Vieira
Coleta	Implícita	Híbrida	Implícita	Implícita
Filtragem	FBC	FC	FBC	Híbrida
Semântica	Não	Não	Sim	Sim
Mineração de dados	Assoc.	Assoc.	Agrupamento	Agrupamento
Validação	Métricas	Uso do sistema	Métricas	Cenários de Uso
Canal de Retorno	Não	Não	Sim	Sim
Middleware	Sim	Sim	Não	Parcialmente

Como avaliado nas Tabelas 9 e 10, os trabalhos diferem principalmente nas técnicas utilizadas e no tipo de coleta de dados. É importante observar que poucos trabalhos utilizam o canal de retorno da TVDi e realizam algum enriquecimento semântico dos dados. Neste sentido, ratifica-se a relevância do trabalho proposto.

Capítulo 7

Considerações Finais

Este trabalho apresentou uma proposta de abordagem baseada em representação do conhecimento para recomendação de conteúdo em TV Digital interativa conectada. Utilizando o *middleware* Ginga como estudo de caso, propôs-se a especificação e disponibilização de um serviço de recomendação que integra conceitos e técnicas da Web Semântica. Este Capítulo apresenta as contribuições deste trabalho de mestrado, as principais dificuldades encontradas durante a pesquisa, bem como uma visão para trabalhos futuros.

7.1 Contribuições

Para este trabalho, foi definida a arquitetura de recomendação de conteúdo no âmbito da plataforma KTV, assim como a formalização semântica da estratégia de recomendação. Adicionalmente foram avaliadas as técnicas e as características do processo de recomendação de trabalhos relacionados.

Adicionalmente foram avaliados os algoritmos de clusterização sobre os dados coletados do usuário e das emissoras de TV. Tal como foi definido o modelo de ontologia que estrutura o conhecimento proveniente da mineração, possibilitando, assim, o raciocínio automático sobre os dados.

Ao final deste trabalho foram obtidas as seguintes contribuições:

1. Avanço no estado da arte em termos dos métodos para recomendação de conteúdo em TVDi conectada;
2. Incorporação ao contexto da TVDi conceitos e técnicas baseadas em conhecimento da Web Semântica, permitindo, assim, o raciocínio

automático sobre as informações, tal como a interoperabilidade de sistemas;

3. Extensão do *middleware* Ginga, permitindo a disponibilização de dados para o serviço de recomendação de conteúdo;
4. Proporcionou o serviço de recomendação de conteúdo no contexto do projeto *KnowledgeTV;*
5. Descrição formal, por meio de uma ontologia, do processo de mineração por tarefas de clusterização;
6. Especificação de uma ontologia para recomendação de conteúdo multimídia, a OntoRKTV;
7. Especificação de regras semânticas sobre a ontologia OntoRKTV.

Há de se destacar que durante o desenvolvimento deste trabalho, inserido dentro do projeto *Knowledge TV*, além dos resultados práticos, também foram alcançados resultados na forma de publicação de trabalhos em eventos científicos, de âmbito nacional e internacional, tais como:

- Lino, N. C; Siebra, C. A.; Vieira, P. K. M.; Ramalho, O. S. Towards the Construction of an Intelligent Platform for Interactive TV.. In: EuroITV 2012, 2012, Berlin. 3rd International Workshop on Future Television Making Television Integrated and Interactive (FutureTV 2012), 2012.
- Vieira, P. K. M.; Lino, N. C. Recomendação de Conteúdo em Ambientes de Convergência Digital Incorporada ao Middleware Ginga. In: XVIII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 12), 2012, São Paulo. WebMedia'12 Brazilian Symposium on Multimedia and the Web São Paulo, Brazil October 15 - 18, 2012, 2012.

7.2 Dificuldades Encontradas

Algumas dificuldades foram enfrentadas durante a definição da proposta e o levantamento do estado da arte em sistemas de recomendação para TV Digital, sendo as principais delas referente ao tipo de técnica de recomendação que a proposta deveria apresentar e qual o cenário para validação da proposta.

Foram encontradas diferentes linhas de recomendação em TV Digital. Foi possível identificar basicamente quatro linhas de investigação na área: (i) atuando na

pesquisa de métodos para identificar com exatidão o usuário que está consumindo, ou assistindo a TV, de forma a buscar-se a mínima ou nenhuma declaração explícita do usuário no processo de identificação; (ii) atuando na pesquisa de métodos para recomendação multiusuário, sem a identificação explícita do usuário, como as técnicas propostas neste trabalho; (iii) pesquisa acerca da integração com outras plataformas e serviços, definindo por exemplo, mecanismos de interação para busca e recuperação de informações por meio de interfaces com a Internet ou do canal de interatividade (enriquecimento semântico das informações); (iv) técnicas de recomendação para oferecer conteúdo, serviços e publicidade personalizada no ambiente da TV, onde são investigadas diversas maneiras para filtragem de dados (FC, FBC, FH).

Outro ponto importante foi a definição da validação da proposta. Dado que é necessário o sistema estar operacional para que possibilite a avaliação da recomendação proposta pelo sistema para cada STB. Dentre os vários trabalhos avaliados, escolhemos realizar uma validação baseada cenários de uso. Desta forma, foram utilizadas bases de dados reais e aplicadas as técnicas do processo proposto, tal como algumas regras para raciocínio automático.

7.3 Trabalhos Futuros

Como trabalho futuro, é importante destacar a necessidade de se implantar algoritmos de diversas tarefas da mineração de dados, além das já analisadas na plataforma *Knowledge TV* (clusterização e associação), que permitirá um comparativo com o RKTV, sendo possível, inclusive, a combinação de técnicas, fornecendo solução para problemas de múltiplas áreas. Também poderá ser estudada a integração da solução do RKTV em *middleware* de outros sistemas de TV digital, tal como em outros ambientes, como a web.

Como forma de enriquecimento e continuação deste trabalho, seria interessante analisar o comportamento do RKTV em relação a outras técnicas de clusterização, tal como outros processos de recomendação, com foco em emissoras, maximizando audiências, ou *marketing*, direcionando produtos de acordo com o público alvo. Ainda neste contexto, seria interessante um estudo aprofundado de técnicas que aperfeiçoassem a identificação do usuário de forma implícita. Tal como um estudo sobre privacidade e segurança dos dados do usuário.

Adicionalmente, destaca-se a importância da especificação e análise de outras regras para raciocínio automático sobre ontologias, além das especificadas neste trabalho, no contexto de recomendação de conteúdo. Tal como a disponibilização da recomendação como um serviço.

Como trabalhos à médio prazo, existem algumas discussões no grupo do projeto KTV acerca de expandir a pesquisa no domínio em que este trabalho foi desenvolvido. Neste sentido, seria considerado o serviço de recomendação de conteúdo em uma segunda tela, como por exemplo, em um dispositivo móvel.

Como trabalho à longo prazo, destaca-se um estudo aprofundado de ontologias no processo de mineração de dados. Expandindo a OntoClusteringKTV, para outras tarefas de clusterização. Especificando um meta serviço que otimize o processo de mineração de dados, indicando, inclusive, o melhor algoritmo a ser utilizado.

Em suma, com a convergência entre estas diferentes áreas abre uma variedade de possibilidades para pesquisa e desenvolvimento de soluções em diferentes graus de complexidade e utilidades.

Referências Bibliográficas

ABNT NBR 15603-1:2007. Televisão Digital Terrestre - Multiplexação e Serviços de Informação (SI) - Parte 1: Serviços de informação do sistema de radiodifusão. 2007

Adomavicius, G. e Tuzhilin, A. Towards the Next Generation of Recommender Systems: A survey of the State-of-the-Art and Possible Extensions. Knowledge and Data Engineering, IEEE Transactions, Volume 17, Issue 6, page(s): 734– 749, June 2005.

Alves, L. G. P.; Kulesza, R.; Silva, F. S.; Jucá, P.; Bressan, G. Análise Comparativa de Metadados em TV Digital. In: II Workshop de TV Digital do Simpósio Brasileiro de Redes de Computadores, 2006, Curitiba. Anais do Simpósio Brasileiro de Redes de Computadores, 2006.

Antoniou, G.; Harmelen, F.; A Semantic Web Primer. 2nd edition. MIT Press, Cambridge Massachusstes, 2008.

Araújo, J. P. C. CoreKTV - Uma infraestrutura baseada em conhecimento para TV Digital Interativa: um estudo de caso para o middleware Ginga. Dissertação de Mestrado. Universidade Federal as Paraíba, 2011.

Ardissono, L.; Gena, C.; Torasso, P.; Bellifemine, F.; Chiarotto, A.; Difino, A.; Negro, B.; *Personalized Recommendation of TV Programs*. 8th AI*IA Conference, Pisa, 2003.

Aroyo, L.; Bellekens, P.; Björkman, M.; Houben, G.; Akkermans, P.; Kaptein, A.; SenSee Framework for Personalized Access to TV Content. European Conference on Interactive TV (EuroITV 2007), 156-165, 2007.

ATSC - Advanced Television System Comitee. Disponível em: <http://www.atsc.org>. Acessado Abril de 2012.

Ávila, P. M. RecommenderTV: Suporte ao Desenvolvimento de Aplicações de Recomendação para o Sistema Brasileiro de TV Digital. Dissertaçao de Mestrado. Universidade Federal de São Carlos, Brasil, 2010.

Babu, G.P. and Murty, M.N. A near-optimal initial seed value selection in K-means algorithm using a genetic algorithm. Pattern Recogn. Lett. 14, 10, 763-169. 1993.

BBC Programmes Ontology. http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml>. Acessado em Agosto de 2012.

Bellekens, P., *Dynamic Integration of Web and TV content for Personalized Information Retrieval in Interactive TV*. master thesis. Department of Mathematics and Computer Science. Eindhoven University of Technology, 2007.

Bennett, J.; Lanning, Stan. The Netflix Prize. In: KDD Cup 2007. 2007.

Berkhin. P.. Survey of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA, 2002.

Berners-Lee, T.; Lassila, O.; Hendler, J. The semantic web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American, 2001.

Berners-Lee. Linked Data - Design Issues. Disponível em: <http://www.w3.org/DesignIssues/LinkedData.html>. Acessado em Abril de 2012.

Berry, M.J.A.; Linoff, G. *Data mining techniques*. John Wiley & Sons, Inc. 1997.

Breitman, K.. Web Semântica: O Futuro da Internet. 1. ed. Rio de Janeiro: LTC - Livros Tecnicos e Cientificos Editora S.A., 2005. v. 1. 190 p.

Bizer, C., Cyganiak, R., and Heath, T. (2007). How to Publish Linked Data on the Web. Disponível em: <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/>. Acessado em Abril de 2012.

Breitman, K.. Web Semântica: O Futuro da Internet. 1. ed. Rio de Janeiro: LTC - Livros Tecnicos e Cientificos Editora S.A., 2005. v. 1. 190 p.

Bettini, C.; Brdiczka, O.; Henricksen, K.; Indulska, J.; Nicklas, D.; Ranganathan, A.; Riboni, D. A Survey of Context Modelling and Reasoning Techniques. In: Journal Pervasive and Mobile Computing. v.6. p161-180. 2010.

Burke, R. Knowledge-based Recommender Systems. 2000.

Burke, R. Hybrid Recommender Systems: survey and experiments. User Modeling and User-Adapted Interaction. Hingham, MA, USA, v.12, n.4, p331-370, 2002.

Kogan, J; Teboulle, M; Nicholas, C; Data Driven Similarity Measures for k-Means Like Clustering Algorithms. Information Retrieval, Volume 8, Issue 2, pp 331-349. 2005.

Chorianopoulos, K.. Personalized and mobile digital TV applications. Proc. 2008 MultimediaTools and Applications, Volume 36, Issue 1-2. Kluwer Academic Publishers, pp. 1-10, 2008.

BDPedia. Disponível em <http://pt.dbpedia.org/>. Acessado em: Janeiro de 2013.

Dentler, K.; Cornet, R.; Teije, A. T.; Keizer, N. Comparison of Reasoners for large Ontologies in the OWL 2 EL Profile. Journal Semantic Web. V 2 Issue 2, pp 71-87. 2011.

Diamantini C., Potena, D. and Storti, E. KDDONTO: an Ontology for Discovery and Composition of KDD Algorithms. In Proc. of the ECML/PKDD09 Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery, pages 13-24, Bled, Slovenia, Sep 7-11 2009.

Dias, M. M. Um modelo de formalização do processo de desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. Florianópolis, 2001.Tese (Doutorado em Engenharia de Produção) – Universidade Federal de Santa Catarina. F. 197.

DTMB – Digital Terrestrial Multimedia Broadcast. Disponível em: <http://www.digitaltvnews.net/content/?cat=410>. Acessado em Abril de 2012.

DVB - Digital Video Broadcast. Disponível em: <http://www.dvb.org>. Acessado em Abril de 2012.

Doan, A.; Madhavan, J.; Domingos, P.; Halevy, A. Ontology Matching: A Machine Learning Approach. Handbook on Ontologies in Information Systems. 2003.

Dogdu, E., Battal, A., A TV Recommendation System Using Semantic Web. ACMSE Conference'10, April 15-17, 2010, Oxford, Missippi, USA.

Équille, L. B. Metadata definition and specification. ENTHRONE Project. 2005.

Evain, J. P.; **Murret-Labarthe**, H.; TV-Anytime Phase 1 – a decisive milestone in open standards for Personal Video Recorders. EBU Technical Review; Julho, 2003.

Fayyad, U. M., PIATESKY-SHAPIRO, G., SMYTH, P. From Data Mining to Knowledge Discovery: an Overview". Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996.

Fensel, D,; Horrocks, i.; Van Harme-len, F.; Decker, S.; Erdmann, M. and Klein. M. Oil in a nutshell. In 12th International Conference on Knowledge Engineering and KnowledgeManagement EKAW2000, Juanles-Pins, France, 2000.

Fensel, D. Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Heidelberg, Germany, 2001.

Fernández, B. Y., Pazos-Arias, J. J., Gil-Solla, A. Ramos-Cabrer, M., López-Nores, M.; AVATAR: An Inproved Solution for Personalized TV based on Semantic Inference; in IEEE Transations on Consumer Electronics, vol. 52, No. 1, p. 223-231, Fev/2006.

Fernández-Lópes, M.; Gómez-Pérez, A. Overview and analysis of methodologies for building ontologies. In: Journal The knowledge Engineering Review. Vol 17, pp 129-156. New York, 2002.

Gómez-Pérez, A. Evaluation of taxonomic knowledge in ontologies and knowledge bases. In Proc. Of the 12th Workshop on Knowledge Acquisition, Modeling and Management, KAW'99. Voyager Inn, Banff, Alberta, Canada. 1999.

Gottgtroy P.; Kasabov, N.; MAcDonell, S..An ontology driven approach for knowledge discovery in Biomedicine. In: Zhang, C., Guesgen, H.W., Yeap, W.K. (Eds.), Lecture notes in artificial intelligence, Vol. 3157. Springer-verlag, Berlin.

Gruber, T. R.; A translation approach to portable ontology specifications. Knowledge Acquisition – 5: 199-220. 1993.

Guarino, N. Formal Ontology, Conceptual Analysis and Knowledge Representation, International Journal of Human-Computer Studies, 43(5-6):625–640, 1995.

Guarino, N.; Formal Ontology and Information Systems. In: Proceedings of FOIS'98. Formal Ontology in Information Systems, Trento. 1998.

Hagah. Disponível em: http://www.hagah.com.br/programacao-tv/jsp/default.jspx?uf=26&action=programacao_operadora&operadora=15. Acessado em Janeiro de 2013.

Hartigan, J. and Wong, M. 1979. Algorithm AS136: A k-means clustering algorithm. Applied Statistics, 28, 100-108.

Han, J. and Kamber, M. Data Mining Concepts and Techniques. 2a Edição, Editora Elsevier, Reino Unido. 2006.

Heath, T. and Bizer, C. Linked Data: Evolving the Web into a Global Data Space. Morgan & Claypool, 1st edition, 2011.

Herlocker, J. L. (2000) "Understanding and Improving Automated Collaborative Filtering Systems", Tese de Doutorado (Doutorado em Ciência da Computação), University of Minnesota, Minnesota.

Hilario, M.; Nguyen, P.; Do, H.; Woznica, A.; Kalousis, A.. Ontology based meta-mining of knowledge discovery workflows. Book chapter in Meta-Learning in Computational Intelligence. Springer, 2011.

Horrocks, i; Connolly, d; Harmelen, f; Mcguinness, d; Patelschneider, p; Stein, l. Daml+oil reference description, 3 2001.

Hsu, S. H.; Wen, M. H.; Lin, H. C.; Lee, C. C.; Lee, C. H: AIMED – *A personalized TV Recommendation System*. In: 5th EUROPEAN CONFERENCE ON INTERACTIVE TELEVISION (EuroITV): INTERACTIVE TV: A SHARED EXPERIENCE, Amsterdam, Netherlands. Proceedings. pp 166-174, 2007.

IBOPE. Disponível em: <www.ibope.com.br>. Acessado em: Janeiro de 2013.

IMDb. Disponível em:<http://www.imdb.com/>. Acessado em: Janeiro de 2013.

ISDB - Integrated Services Digital Broadcasting. Disponível em: <http://www.dibeg.org>. Acessado em Abril 2012.

ITU. ITU-T Recommendation J.200: Worldwide common core – Application environment for digital interactive television services. 2001.

IBGE - Instituto Brasileiro de Geografia e Estatística. Síntese de Indicadores Sociais. Brasil, 2012.

Jain, A. K., Murty, M. N., and Flynn, P. J.. Data clustering: a review. ACM Comput. Surv, 1999. 31(3):264–323.

Java. Disponível em: <http://www.java.com/pt_BR/>. Acessado em Janeiro de 2013.

Jena. Disponível em: < http://jena.sourceforge.net/>. Acessado em Janeiro de 2013.

Jones, G. A., Defilippis, J. M., Hoffman, H., e Williams, E. A.. Digital Television Station and Network Implementation. Proceedings of the IEEE, 94 (1), pp. 22-36. 2006.

Khalilian, M., Mustapha, N., Suliman, N. , Mamat, A. A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets. IMECS, 2010. Hong Kong.

Kim, J., Kwon, E., Cho, Y., Kang, S. Recommendation System of IPTV TV Program Using Ontology and K-means Clustering. Second International Conference, UCMA 2011, Daejeon, Korea, April 13-15, 2011. Proceedings, Part II.

Kimball, R. Data Warehouse Toolkit. São Paulo: Makron Books. 1998.

Kononenko, I. & Kukar, M. machin learning and data mining. Chichester, UK: Horwood Publishing, 2007.

Lécué, F., Combining Collaborative Filtering and Semantic Content-based Approaches to Recommend Web Services. ICSC, pp.200-205, 2010 IEEE Fourth International Conference on Semantic Computing, 2010

Leite, L. E. C., et al. FlexTV – Uma Proposta de Arquitetura de Middleware para o Sistema Brasileiro de TV Digital. Revista de Engenharia de Computação e Sistemas Digitais. Vol. 2, pp. 29-50. 2005.

Linden, G., B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item

Collaborative Filtering. IEEE Internet Computing, Jan.-Feb. 2003.

Lino, N., Araújo., Anakubi, D., Júnior, J., Martins, M., Nóbrega, R., Amaro, M., Siebra, C., Lemos G. Knowledge TV. EuroITV 2011(Lisboa).

LOD. W3C SWEO Community Project <http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>. Acessado em Abril de 2012.

Lugmayr, A.; NIIRANEN, S.; KALLI, S.. Digital Interactive TV and Metadata. Springer, 2004.

Matlab. Discponível em:<http://www.mathworks.com/products/matlab/>. Acessado em Maio de 2013.

Marques, M. C., Comparação entre os métodos de agrupamentos K-Means de Kohonen em análise de pesquisa de mercado. Revista ICA – Inteligência Computacional Aplicada nº1. 2008.

Martín, A., Celestino, S., Valdenebro, A., Mensaque, J.. Ontologias e Inteligencia Artificial para La Recuperación Eficiente Del Conocimiento. *XV* Jornadas Bibliotecarias de Andalucía. Octubre, 2009

Médola, A. S. L.. Televisão Digital Brasileira e os Novos Processos de Produção de Conteúdos: Os Desafios para o Comunicador. Revista da Associação Nacional dos Programas de Pós-Graduação em Comunicação | E-compós, Brasília, v.12, n.3, set./dez. 2009.

Middleton, S., *Capturing Knowledge of User Preferences with Recommender Systems*, tese de doutorado. Department of Electronic and Computer Science. University of Southampton, 2003.

Miranda, D. S. ; Azevedo, L. L. S. ; Magalhães, R. P. . Consumindo Linked Data na Web. In: Encontro Unificado de Computação em Parnaíba, 2011, Parnaíba. Consumindo Linked Data na Web, 2011.

Montaner, M. Collaborative Recommender Agents Based on Case-Based Reasoning and Trust. 2003.

Montez, C.; Becker, V. TV Digital Interativa: Conceitos, Desafios e Perspectivas para o Brasil. Ed. da UFSC, 2ª Edição, Florianópolis, Brasil, 2005.

MPEG-2 - ISO/IEC 13818-1.2008. Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems. 1992.

NetFlix. Disponível em <www.netflix.com>. Acessado em Janeiro de 2013.

Neto, M. M., Cardoso, D., Teixeira, C. T., Cortés, M. Abordagem Combinada para Recomendação Personalizada Utilizando o Guia de Programação Eletrônico. CSBC' 2010, Belo Horizonte, MG, 2010.

Nigro, H. O.; Cisaro S. G.; Xodo, D. H. Data Mining With Ontologies: Implementations, Findings and Frameworks, Information. Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2007.

Noy, F. N.; McGuinness, D. L. Ontology Development 101: A Guide to Creating Your First Ontology'. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

Ochi, L. S., Dias, C. R., Stênio, S. F.. Clusterização em Mineração de Dados. *Escola Regional de Informática Rio de Janeiro*. Espírito Santo. Mini Curso. Rio das Ostras, 2004.

OWL. Disponível em: <http://www.w3.org/OWL/>. Acessado em Novembro de 2012.

Patrício Júnior, J. C. A. Mining Knowledge TV: Uma Abordagem de Ambiente de KDD com Ênfase em Mineração de Dados no Ambiente da Knowledge TV. Dissertação de Mestrado. Universidade Federal as Paraíba, 2012.

Piccioni, C. A. Modelo e Implementação de um Serviço de Datacasting para Televisão Digital. Dissertação de Mestrado. 102p. Universidade Federal de Santa Catarina. Florianópolis, 2005.

PostgreSQL. Disponível em: < http://www.postgresql.org/>. Acessado em Janeiro de 2013.

Protegé. Disponível em: <http://protege.stanford.edu/>. Acessado em Janeiro de 2013.

RDF. Disponível em: <http://www.w3.org/RDF/>. Acessado em Abril de 2012.

Reateguil, E. ; **Cazella**, S. C.. Minicurso: Sistemas de Recomendação. In: ENIA Encontro Nacional de Inteligência Artificial, 2005, São Leopoldo, Sociedade Brasileira de Computação. 2005.

Resnick, P. and Varian, H. R. 1997. Recommender systems. Communications of the ACM. ACM 40, 3 (Mar. 1997), 56-58.

Russell, S. and Norvig, P. Artificial Intelligence - A Modern Approach. 2 edição, Prentice Hall, 2002.

Sallton, G.; Buckley, C. Term weighting approaches in automatic text retrieval. Information Processing and Management, Vol. 5, p. 513-523, 1988.

Sarwar B. M.; Karipys, G.; Konstan, J, A.; Reidl, J. Item-based colaborative filtering recommendation algorithms. In Word Wide Web, 2001.

SBTVD – Sistema Brasileiro de TV Digital. Ministério das Comunicações. Disponível em: <http://sbtvd.cpqd.com.br/>. Acessado em Abril de 2012.

Schafer, J. B.: MetaLens: A Framework for Multi-source Recommendations. Tese em Ciência da Computação, University of Minnesota, 2001.

Silva. C. F, Lino, N. C. Q, Vieira, P.K.M. Semantic Integration - Uma Extensão do Núcleo do Middleware Ginga. Simpósio Brasileiro de Sistemas de Informação. 2013.

Silva, M. P. S. Mineração de Dados: Conceitos, Aplicações e Experimentos com Weka. Livro da Escola Regional de Informática Rio de Janeiro - Espírito Santo. Porto Alegre: Sociedade Brasileira de Computação, v. 1, p. 1-20, 2004.

Silva, E. P.. Classificação de Informação Baseada em Ontologias. Dissertação de Mestrado. Universidade Federal de Alagoas (UFAL). Maceió, 2006.

Silva, R. R., Ferreira, M. G. V., Vijaykumar, N. L.. Uma Ontologia Baseada em um Meta-Modelo Orientado a Objetos para Descrição de Domínios e Problemas de Planejamento da Área Espacial. II Seminário de Pesquisa em Ontologia no Brasil. Rio de Janeiro, 2009.

Sky. Disponível em: <http://www.sky.com.br/servicos/GuiadaTv/GravacaoDistancia.aspx>. Acessado em Janeiro de 2013.

Soares, L. F.; Lemos, G. Interactive Television in Brazil: System Software and the Digital Divide. In European Interactive TV Conference - EuroITV2007. Amsterdam, 2007.

Souza Filho, G. L. de, Leite, L. E. C., Batista, C. E. C. F. Ginga-J: The Procedural Middleware for the Brazilian Digital TV System. In: Journal of the Brazilian Computer Society, no 4, Vol 12, (ISSN 0104-6500) pp. 47-56. Março, 2007. Porto Alegre, RS, Brasil.

Tan, P.N., Steinbach, M., Kumar, V. *Introduction to Data Mining*, Addison-Wesley, 2006.

Truyen, T.T.; Phung, D.Q.; Venkatesh, S.; *Preference Networks: Probabilistic Models for Recommendation Systems*, 6th Australasian Data Mining Conference (AusDM 2007), 2007.

Torres, R. Personalização na Internet. Editora Novatec, 2004.

TVA. TV-Anytime Forum, Disponível em: <http://www.tv-anytime.org>. Acessado em Janeiro de 2013.

TV Arapuã. Disponível em: <http://www.tvarapuan.com.br/>. Acessado em: Jeneiro de 2013.

Uschold, M. Building ontologies: towards a unified methodology. In: Annual Conference of the British Computer Society Specialist Group of Expert Systems, 16., 1996, Cambridge, UK. p.1-17. 1996.

XML. Extensible Markup Language. Disponível em: < http://www.w3.org/XML/>. Acessado em Janeiro de 2013.

Xu, J.; Zhang, L.; LU, H.; LI, Y. The Development and Prospect of Personalized TV Program, In: Proceedings of the IEEE 4th International Symposium on Multimedia Software Engineering (MSE'02), 2002.

Wang, H.; Nearest Neighbor by Neighborhood Counting. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.28, no.6, pp. 942-953. 2006.

Weka. Disponível em: < http://www.cs.waikato.ac.nz/ml/weka/>. Acessado em Janeiro de 2013.

W3C. World Wide Web Consortium. W3C Semantic Web Activity. Disponível em: < http://www.w3.org/2001/sw/>. Acesso em Junho de 2012.

APÊNDICE A – Clusterização dos dados

```
base = load('filmes.mat');

base = base.filmes;

ids = unique(base(:,1));

result = [];
for k = 1:length(ids)
    w = sum(base(base(:,1)==ids(k),3:size(base,2)),1);
    result = [result; [ids(k), w]];
end

data = result(:,2:21);

for k = 2:300

  [idx4,cent4,sumdist] =
kmeans(data,k,'distance','sqEuclidea','EmptyAction', 'singleton',
'Replicates', 5);
  graph2(k-1)= sum(sumdist);
  a=silhouette(data,idx4,'sqEuclidea');
  graph(k-1) = mean(a);
end

plot (2:300,graph2);
figure
plot(2:300,graph);
new_data = [idx4,result];
```

APÊNDICE B – Implementações RKTV

```
import java.util.ArrayList;


public class User {
String user_id;
Historico historico;
ArrayList<User> vizinhos;
ArrayList<String> recomendacoes;

public User(String user_id) {

	this.user_id = user_id;
	this.historico = new Historico();
	this.vizinhos = new ArrayList<User>();
	this.recomendacoes = new ArrayList<String>();
}

public ArrayList<User> getVizinhos() {
	return vizinhos;
}


public ArrayList<String> getRecomendacoes(){
	return this.recomendacoes;
}

public void insertVizinho (User user){
	this.vizinhos.add(user);
}

public void setRecomendacao (ArrayList<String> recomendacoes){
	this.recomendacoes = recomendacoes;

}

public void insertRecomendacao (String recomendacao){
	this.recomendacoes.add(recomendacao);

}

public void setVizinhos(ArrayList<User> vizinhos) {
	this.vizinhos = vizinhos;
```

```
}

public String getUser_id() {
	return user_id;
}

public void setUser_id(String user_id) {
	this.user_id = user_id;
}

public Historico getHistorico() {
	return historico;
}

public void setHistorico(Historico historico) {
	this.historico = historico;
}

}


import java.util.ArrayList;

public class Recomendacao {

	public void RecomendacaoPorUsuario(User user_id) {
		ArrayList<String> filmesAseremRecomendados = new
ArrayList<String>();


		for (int i = 0; i < user_id.getHistorico().getFilmes().size(); i++)
{

	System.out.println(user_id.getHistorico().getFilmes().get(i));

		}

		for (int i = 0; i < user_id.getVizinhos().size(); i++) {
			for (int j = 0; j <
user_id.getVizinhos().get(i).getHistorico().filmes
					.size(); j++) {

				System.out.println("user Id " +
user_id.getVizinhos().get(i).getUser_id() + " "+ user_id.getVizinhos().get(i)
					.getHistorico().filmes.get(j)+ " id filme
visto");
				if
(!(filmesAseremRecomendados.contains(user_id.getVizinhos()
					.get(i).getHistorico().filmes.get(j)))) {

	filmesAseremRecomendados.add(user_id.getVizinhos().get(i)
						.getHistorico().filmes.get(j));

				}

			}

		}
```

```
			filmesAseremRecomendados.removeAll(user_id.getHistorico().filmes);
			user_id.setRecomendacao(filmesAseremRecomendados);



	}
	public void recomendacaoTodoSistema(Sistema sistema){


			for (int i = 0; i < sistema.getSistema().size(); i++) {
				for (int j = 0; j <
sistema.getSistema().get(i).getUsersId().size(); j++) {

	RecomendacaoPorUsuario(sistema.getSistema().get(i).getUsersId().get(j));

				}

			}

	}

}
import java.util.ArrayList;


public class Cluster {
	public int clusterId;
	public ArrayList<User> usersId;
	public Cluster(int clusterId, ArrayList<User> usersId) {
			super();
			this.clusterId = clusterId;
			this.usersId = usersId;
	}
	public int getClusterId() {
			return clusterId;
	}
	public void setClusterId(int clusterId) {
			this.clusterId = clusterId;
	}
	public ArrayList<User> getUsersId() {
			return usersId;
	}
	public void setUsersId(ArrayList<User> usersId) {
			this.usersId = usersId;
	}
	public User getUser (String userID){
			for (int i = 0; i < this.usersId.size(); i++) {
				if (usersId.get(i).getUser_id().equals(userID)) {
					//System.out.println("teste2");
					return usersId.get(i);

				}

			}
			return null;
```

```
	}


}

import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.util.ArrayList;

public class Conexao {
	String url, usuario, senha;
	Connection con;
	ResultSet resultset;

	public Conexao() {

		try {

			this.url = "jdbc:postgresql://localhost:5432/NetFlix";
			this.usuario = "postgres";
			this.senha = "bd";

			Class.forName("org.postgresql.Driver");

			con = DriverManager.getConnection(url, usuario, senha);

		} catch (Exception e) {
			e.printStackTrace();
		}
	}


	public float  consultaPrecisaoRecomendacaoPorUser(User user_id)
			throws SQLException {
		float precisaoPorUser = 0;
		float TotalAcerto = 0;
		Statement stm;
		stm = con.createStatement();

		ResultSet resultset = stm
				.executeQuery("SELECT   count(*) as totalAcertos FROM
ratings where movie_id in ("
					+
formataFilmes(user_id.getRecomendacoes())
					+ " ) and user_id = " +
user_id.getUser_id());

		while (resultset.next()) {
			TotalAcerto = (int) resultset.getInt("totalAcertos");

		}

		return precisaoPorUser = TotalAcerto /
user_id.getRecomendacoes().size();
```

```
	}


	public void consultaPrecisaoRecomendacaoSistema(Sistema sistema) throws
SQLException{
		float precisaoTotal = 0;
		float totalUserAvaliados = 0;

		for (int i = 0; i < sistema.getSistema().size(); i++) {
			for (int j = 0; j <
sistema.getSistema().get(i).getUsersId().size(); j++) {
				if
(sistema.getSistema().get(i).getUsersId().get(j).recomendacoes.size()!= 0) {
					totalUserAvaliados++;
					precisaoTotal =
precisaoTotal+consultaPrecisaoRecomendacaoPorUser(sistema.getSistema().get(i).ge
tUsersId().get(j));

				}

			}

		}

	}

	public ResultSet getResultset() {
		return resultset;
	}

	public void setResultset(ResultSet resultset) {
		this.resultset = resultset;
	}

		public String formataFilmes(ArrayList<String> lista) {
		String saida = "";
		for (int i = 0; i < lista.size(); i++) {
			saida = saida + lista.get(i) + ",";

		}
		String retorno = saida.substring(0, saida.length() - 1);

		return retorno;

	}

	public String formataUsersID(ArrayList<User> lista) {
		String saida = "";
		for (int i = 0; i < lista.size(); i++) {
			saida = saida + lista.get(i).getUser_id() + ",";

		}
		// System.out.println(lista.size() + "ver");
		String retorno = saida.substring(0, saida.length() - 1);

		return retorno;
```

```
	}

	public void CriaRede(Sistema sistema) throws SQLException {
		for (int i = 0; i < sistema.getSistema().size(); i++) {
			for (int j = 0; j <
sistema.getSistema().get(i).usersId.size(); j++) {


	System.out.println(sistema.getSistema().get(i).usersId.size()
					+ " tamanho cluster ");
				if (sistema.getSistema().get(i).usersId.size() > 1) {

	System.out.println(sistema.getSistema().get(i).usersId
					.size() + " tamanho cluster");

				ArrayList<User> t =
consultaUSersComaMaiorSimilaridade(
					sistema.getSistema().get(i),
sistema.getSistema()

	.get(i).usersId.get(j));

				System.out.println(t.size()
					+ " tamanho array vizinhos"
					+
sistema.getSistema().get(i).getClusterId()
					+ " cluster Id"
					+
sistema.getSistema().get(i).usersId.get(j)
							.getUser_id() + " user
id");

	sistema.getSistema().get(i).usersId.get(j).setVizinhos(t);

				}
			}

		}

	}

	public void InicializaHistoricoPorusuario(Sistema sistema)
			throws SQLException {

		for (int i = 0; i < sistema.getSistema().size(); i++) {
			for (int j = 0; j <
sistema.getSistema().get(i).getUsersId().size(); j++) {
				Statement stm;
				stm = con.createStatement();
				ResultSet resultset = stm
					.executeQuery("Select * FROM
limitefilmedel where user_id ="
						+
sistema.getSistema().get(i).getUsersId()

	.get(j).getUser_id() + "and rate>=4");
				while (resultset.next()) {
					sistema.getSistema().get(i).getUsersId().get(j)
```

```
                                              .getHistorico()

      .insertFilme(resultset.getString("movie_id"));

                        }

                  }

            }

      }

      public void consultaHistoricoPorusuario(Cluster cluster, User user_id)
                  throws SQLException {
            Statement stm;
            stm = con.createStatement();
            ResultSet resultset = stm
                        .executeQuery("Select * FROM limitefilmedel where
user_id ="
                                    + user_id.getUser_id() + "and rate>=4");
            while (resultset.next()) {

      user_id.getHistorico().insertFilme(resultset.getString("movie_id"));

            }
      }

      public ArrayList<User> consultaUSersComaMaiorSimilaridade(Cluster
cluster,
                  User userID) throws SQLException {
            Statement stm;
            stm = con.createStatement();
            ArrayList<User> clusterAux = (ArrayList<User>) cluster.getUsersId()
                        .clone();
            for (int i = 0; i < clusterAux.size(); i++) {
                  if (clusterAux.get(i).user_id.equals(userID.getUser_id())) {
                        clusterAux.remove(i);
                  }

            }

            ResultSet resultset = stm
                        .executeQuery("SELECT  user_id, count(*) FROM
limitefilmedel where movie_id in ("
                                    +
formataFilmes(userID.getHistorico().getFilmes())
                                    + " ) and user_id in ("
                                    + formataUsersID(clusterAux)
                                    + " )and rate>=4"
                                    + "group by user_id having count(*)>= all
(select count(*) from limitefilmedel where movie_id in ("
                                    +
formataFilmes(userID.getHistorico().getFilmes())
                                    + " )and user_id in ("
                                    + formataUsersID(clusterAux)
                                    + " )and rate>=4 group by user_id)");
            ArrayList<String> usuariosMaisSimilares = new ArrayList<String>();
```

```
            while (resultset.next()) {
                  usuariosMaisSimilares.add(resultset.getString("user_id"));

            }
            ArrayList<User> retorno = addVizinhos(usuariosMaisSimilares,
userID,
                        cluster);
            return retorno;

      }

      public ArrayList<User> addVizinhos(ArrayList<String> vizinhos, User user,
                  Cluster cluster) {
            ArrayList<User> vizinhosAdd = new ArrayList<User>();

            for (int i = 0; i < vizinhos.size(); i++) {

                  vizinhosAdd.add(cluster.getUser(vizinhos.get(i)));

            }

            return vizinhosAdd;
      }

      public String getUrl() {
            return url;
      }

      public void setUrl(String url) {
            this.url = url;
      }

      public String getUsuario() {
            return usuario;
      }

      public void setUsuario(String usuario) {
            this.usuario = usuario;
      }

      public String getSenha() {
            return senha;
      }

      public void setSenha(String senha) {
            this.senha = senha;
      }

      public Connection getCon() {
            return con;
      }

      public void setCon(Connection con) {
            this.con = con;
      }
}
```