

TÓPICOS EM BANCO DE DADOS, MULTIMÍDIA E WEB

I EDIÇÃO DOS MINICURSOS CONJUNTOS DO SBBD E WEBMEDIA

XXVI Simpósio Brasileiro de Banco de Dados (SBBD)

XVII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)

03 de Outubro de 2011

Florianópolis - Santa Catarina

TÓPICOS EM BANCO DE DADOS, MULTIMÍDIA E WEB

Editora

Sociedade Brasileira de Computação

Organizadores

Patrícia Vilain (UFSC)

Valter Roesler (UFRGS)

Realização

Universidade Federal de Santa Catarina - UFSC

Promoção

Sociedade Brasileira de Computação

Ficha catalográfica elaborada na Biblioteca Central da
Universidade Federal de Santa Catarina

S612t Simpósio Brasileiro de Sistemas Multimídia e Web (17. : 2011 : Florianópolis, SC).
Tópicos em banco de dados, multimídia e Web / XVII Simpósio Brasileiro de Sistemas Multimídia e Web, XXVI Simpósio Brasileiro de Banco de Dados ; organizadores Patrícia Vilain, Valter Roesler. – Florianópolis : Sociedade Brasileira de Computação, 2011.
188 p.

1. edição dos minicursos conjuntos do SBBD e WebMedia 2011.
Realização: Universidade Federal de Santa Catarina
ISBN:

1.Sistemas multimídia – Congressos. 2. World Wide Web (Sistema de recuperação da informação). 3. Banco de dados. I. Simpósio Brasileiro de Banco de Dados (26. : 2011 : Florianópolis, SC) II. Vilain, Patrícia. III. Roesler, Valter. IV. Sociedade Brasileira de Computação. III. Título.
CDU: 681.31.022(84)

Prefácio

Este livro é a coletânea de textos fornecidos pelos autores dos minicursos que foram apresentados durante o XXVI Simpósio Brasileiro de Bancos de Dados (SBBDD 2011) e o XVII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2011), ocorridos em Outubro de 2011 em Florianópolis. Estes minicursos são eventos de curta duração (4 horas) que visam apresentar uma visão de tópicos atuais de pesquisa e/ou tecnologia de interesse das comunidades de Banco de Dados ou de Sistemas Multimídia e Web. Cada minicurso refere-se a um tema atual e de interesse dos ouvintes, oferecendo-lhes uma oportunidade de aprenderem sobre aquele tema e também de utilizarem esse novo conhecimento em suas áreas de atuação ou pesquisa.

Os minicursos selecionados para o SBBDD 2011 e WebMedia 2011 foram submetidos a uma chamada pública divulgada através da lista eletrônica da SBC (Sociedade Brasileira de Computação). Foram recebidas 18 propostas de minicursos que foram avaliadas pelo comitê do SBBDD, composto pelas professoras Fernanda Baião, Patrícia Vilain e Renata Galante, e pelo comitê do WebMedia, composto pelos professores Eduardo Barrére, Maria da Graça Campos Pimentel e Valter Roesler. Durante a avaliação foram consideradas a relevância para o evento, expectativa do público, atualidade e conteúdo de cada minicurso. Ao final, foram selecionadas 6 propostas, às quais se referem os capítulos deste livro.

O Capítulo 1 apresenta o texto do minicurso intitulado **Cloud computing: elástica e segura**. Computação nas nuvens (*cloud computing*) consiste de um conjunto de tecnologias que provê aos seus usuários: gerenciamento, uso sob demanda, adequação às necessidades, racionalização do uso dos recursos e automação dos processos relacionados à criação de infra-estruturas de suporte. Este minicurso apresenta como as questões de performance e segurança devem ser tratadas para que aplicações existentes ou em desenvolvimento possam ser migradas para a nuvem.

O Capítulo 2 apresenta o texto do minicurso intitulado **Interfaces para aplicações de Interação Natural baseadas na API OpenNI e na Plataforma Kinect**. Interfaces em linguagem natural estão se popularizando, e um dos motivos para isso é a diminuição do custo dos equipamentos de sensoriamento e processamento de imagens. Em particular, a plataforma Kinect se popularizou no Xbox 360, entretanto, a API foi liberada e pode ser utilizada em diferentes sistemas operacionais. Em paralelo, várias aplicações surgiram utilizando tais interfaces, indo desde a área de jogos até aplicações em medicina. O curso apresenta os mecanismos de desenvolvimento para plataformas em linguagem natural, especificamente focando no Kinect, que é uma das mais populares.

O Capítulo 3 apresenta o texto do minicurso intitulado **Linked Data: Construindo um Espaço de Dados Global na Web**. *Linked Data* define um conjunto de princípios que formam a base para a difusão e uso de dados na Web. Este minicurso provê uma base conceitual e prática relacionada à *Linked Data*, onde são apresentados os seus fundamentos e discutidas ferramentas para publicação e consumo de dados na Web. Também são discutidos o estado da arte, as limitações, os desafios e questões em aberto desta área de pesquisa.

O Capítulo 4 apresenta o texto do minicurso intitulado **O que sua Personalidade revela? Fidelizando clientes web através de Sistemas de Recomendação e Traços de Personalidade**. O minicurso apresenta a importância de levar em consideração os aspectos psicológicos humanos, tais como a emoção e traços de personalidade, para criar uma forma de comunicação humano-computador mais efetiva. Assim, o computador pode levar em consideração o entendimento da natureza psicológica humana e procurar se adaptar ao ser humano visando uma interação personalizada. O minicurso busca efetivamente melhorar o entendimento de tais níveis de interação, e o participante, com a base oferecida pelo curso, terá uma visão diferenciada dos mecanismos mais eficientes para personalizar a interação com seu público.

O Capítulo 5 apresenta o texto do minicurso intitulado **Por dentro das redes complexas: detectando grupos e prevendo ligações**. A área de mineração de redes complexas é uma área nova e multidisciplinar onde destacam-se as tarefas de extração de propriedades estatísticas, a detecção de comunidades e a predição de ligações. Este minicurso introduz os conceitos básicos da mineração de redes complexas, por exemplo, a WWW e redes sociais como Orkut e Facebook, além de técnicas de detecção de comunidade e predição de ligação.

O Capítulo 6 apresenta o texto do minicurso intitulado **Utilizando a Arthron para o Gerenciamento Remoto de Múltiplos Fluxos de Mídia**. Múltiplos fluxos de vídeo são utilizados atualmente em diversas aplicações, indo desde a área de eventos artísticos e esportivos, até áreas de telemedicina. O minicurso apresenta a ferramenta Arthron, que consiste em um conjunto de componentes que permite a gerência e controle de tais fluxos com uma interface fácil e intuitiva, indo desde a etapa de captura até a exibição das mídias envolvidas. Além disso, a ferramenta Arthron trabalha com diferentes qualidades de vídeo, podendo se adaptar às condições de infra-estrutura existentes.

Esperamos que este livro seja útil para todos aqueles interessados e praticantes das áreas de Banco de Dados e Sistemas Multimídia e Web.

Patrícia Vilain (UFSC) e Valter Roesler (UFRGS)
Coordenadores dos Minicursos do SBBB/WebMedia 2011

Sumário

Capítulo 1

Cloud Computing: Elástica e Segura..... 1

Capítulo 2

Interfaces para Aplicações de Interação Natural Baseadas na API OpenNI e na
Plataforma Kinect 35

Capítulo 3

Linked Data: Construindo um Espaço de Dados Global na Web 61

Capítulo 4

O que sua Personalidade Revela? Fidelizando Clientes Web através de Sistemas
de Recomendação e Traços de Personalidade 91

Capítulo 5

Por Dentro das Redes Complexas: Detectando Grupos e Prevendo Ligações 123

Capítulo 6

Utilizando a Arthron para o Gerenciamento Remoto de Múltiplos Fluxos de
Mídia..... 149

Capítulo

1

Cloud computing: elásticas e seguras

Rodrigo Elia Assad^{1,2}, Vinicius de Melo Rocha^{1,2}, Emanuell Faustino²,
Felipe Silva Ferraz^{1,2} e Silvio Romeiro Lemos Meira¹

¹C.E.S.A.R - Centro de Estudos e Sistemas Avançados do Recife {assad@cesar.org.br, vmr@cesar.org.br, Felipe.ferraz@cesar.org.br }

²Universidade Federal de Pernambuco {rea@cin.ufpe.br, vmr@cin.ufpe.br, efhl@cin.ufpe.br, fsf3@cin.ufpe.br, srlm@cin.ufpe.br }

Abstract

With increasing connectivity speed and Web systems evolution, emerges the Internet systems, which are more commonly called cloud computing. It's designates a support platform that provides: management, on-demand use, fitness requirements, rational use of resources and automation of processes related to creation of infrastructure and support that should be secure .

Resumo

Com o aumento da velocidade de conectividade e evolução dos sistemas WEB, começa a surgir os sistemas de Internet, que mais comumente são chamados de Computação nas Nuvens. Este termo designa uma plataforma de suporte a sistemas de software que provê aos seus usuários: gerenciamento, uso sob demanda, adequação às necessidades, racionalização do uso dos recursos e automação dos processos relacionados a criação de infra-estruturas de suporte e que devem ser seguros.

1.1 Introdução a Cloud Computing

Atualmente, Computação nas Nuvens (*Cloud Computing*) tem sido um dos temas mais discutidos na área da Ciência da Computação, tanto como tópico de pesquisa, como de interesse de investimento para as empresas. Os conceitos associados à Computação nas nuvens começaram a ter uma maior visibilidade em meados do ano de 2008, fundamentando-se em 2009. Como fatores que impulsionaram a disseminação destes conceitos e idéias associadas, tem-se a crise econômica e a necessidade de racionalização dos custos de tecnologia da informação e comunicação (TIC).

Computação nas nuvens pode ser vista como uma plataforma de suporte a sistemas de software que provê aos seus usuários: gerenciamento, uso sobre demanda, adequação as

necessidades, racionalização do uso dos recursos e automação dos processos relacionados a criação de infra-estruturas de suporte.

A idéia central do uso de computação nas nuvens está em tornar a infra-estrutura de suporte às aplicações o mais programável possível. Este novo paradigma faz com que os usuários/clientes das empresas que provêm serviços em *clouds* paguem somente pelo consumo real de recursos, sendo os mesmos disponibilizados de imediato a medida que precisem de mais capacidade.

Abaixo estão listados os principais benefícios obtidos com o uso de Computação nas Nuvens

- Agilidade: permite aos usuários crescer a sua infra-estrutura de forma rápida;
- Custo: diminuição dos custos, uma vez que os usuários passam a pagar pelo que utilizam e não mais por toda a infra-estrutura;
- Independência de dispositivos e localização: os usuários passarão a utilizar um *browser* para acessar os seus recursos;
- Compartilhamento: como a idéia básica é que os usuários paguem apenas pelo que utilizem, logo o compartilhamento dos recursos computacionais se faz necessário;
- Disponibilidade: os sistemas precisam estar disponíveis a qualquer momento, logo é necessário redundância;
- Escalabilidade: os sistemas devem possuir escalabilidade horizontal, ou seja, a capacidade de se expandir apenas adicionando mais servidores.
- Segurança: devido a centralização dos sistemas, fica mais fácil implementar questões de segurança
- Manutenção: por não ser necessário instalar os aplicativos nos computadores dos usuários, fica mais fácil realizar a manutenção e atualização dos sistemas.
- Medição: com a centralização o monitoramento torna-se mais fácil, ajudando na tomada de decisão de quando se deve ou não adquirir mais infra-estrutura.

Como se pode observar, estes benefícios são interessantes do ponto de vista de quem desenvolve os sistemas, de quem os administra e de quem os mantém. Eles permitem que novas tecnologias sejam adicionadas ao dia a dia da empresa sem a necessidade de um investimento elevado.

Dentre as tecnologias que dão suporte ao crescimento elástico dos sistemas computacionais na nuvem, destacam-se: virtualização, *peer-to-peer* e grids computacionais. A junção destas tecnologias associado a estratégias de gerenciamento, segurança e a compreensão básica do funcionamento das aplicações, permitem a criação de um arcabouço de sustentação para os modelos de solução descritos acima, tendo como desafio a exigência de prover todas elas de forma segura.

1.1.1 Principais benefícios

Além da redução de custos, *Cloud Computing* provê aos seus usuários: gerenciamento, uso sobre demanda, adequação as necessidades, racionalização do uso dos recursos e automação dos processos relacionados a criação de infra-estruturas de suporte.

1.2 Sistemas P2P

1.2.1 Arquiteturas P2P

Os grandes responsáveis pelo impulso e popularização dos sistemas distribuídos P2P foram o Napster [13] e Gnutella [14]. Além de protagonizar inúmeras questões judiciais quanto a direitos autorais e pirataria, foi através destas ferramentas que se evidenciou o potencial da tecnologia no que diz respeito a compartilhamento de recursos sem desprender maiores investimentos em hardware. Desde a época desses precursores, a tecnologia P2P vem sofrendo diversas mudanças. Esse tipo de sistema computacional encontra-se classificado dentre alguns tipos de sistemas e com algumas subdivisões, como pode ser observado na Figura 1.

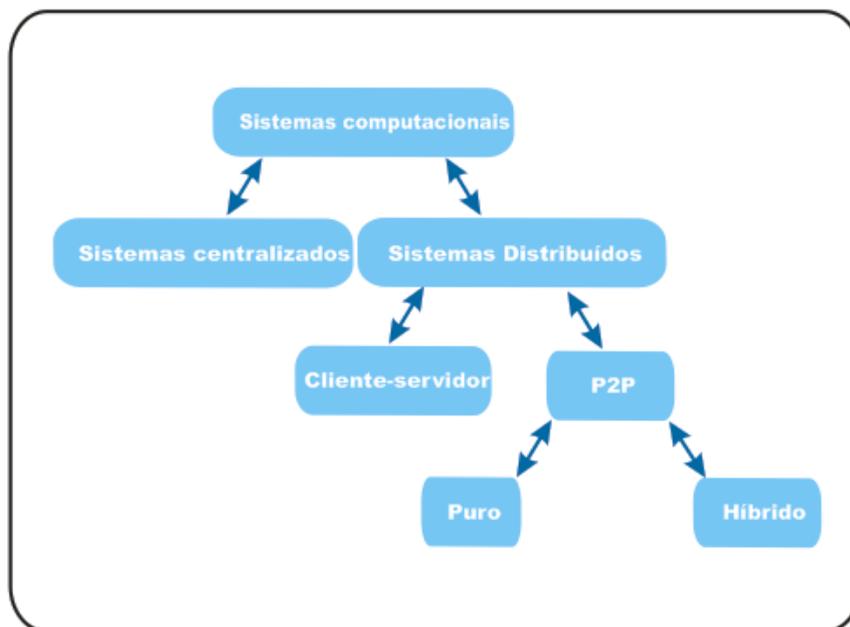


Figura 1 Taxonomia de sistemas P2P

A ilustração mostra que os sistemas tendem a sofrer um processo de descentralização contínuo, onde o primeiro grande avanço se deu com o modelo cliente-servidor. Este por sua vez, consiste numa máquina central que disponibiliza algum tipo de serviço consumido por outras máquinas. A segunda forma de descentralização já mostra o surgimento das aplicações P2P, classificadas por sua forma completamente descentralizada (ou pura) e um modelo híbrido. É comum nos sistemas P2P híbridos a existência de estruturas de controle centralizadas para a utilização de recursos descentralizados. Cada um dos modelos de descentralização possuem suas vantagens e desvantagens. Essas características são abordadas nas seções seguintes.

1.2.2 Arquiteturas puras

As redes denominadas puras possuem como principal característica a não existência de controle central. Todo o funcionamento se dá com uso de um algoritmo descentralizado onde é possível localizar *peers* e/ou serviços [58].

Esta localização é feita fazendo uso da técnica de enchente (*flooding*)[58], onde a mensagem é enviada a todos os computadores diretamente ligados ao emissor e cada máquina que recebe a mensagem faz o mesmo. Para evitar que a rede entre em colapso (*loop*), um tempo

de vida é atribuído à mensagem para os casos em que ela não chegue ao seu destino. Os pontos negativos aqui são:

- Algumas máquinas podem não receber a mensagem, negando assim um serviço que estaria disponível em tese;
- Há um grande volume de mensagens enviadas até que a mesma encontre a máquina de destino.

Para melhor ilustrar o funcionamento, é possível visualizar na Figura 2 o algoritmo de busca em execução. É possível perceber que algumas máquinas não repassam a mensagem recebida devido a limitações no tempo de vida e número de repasse.

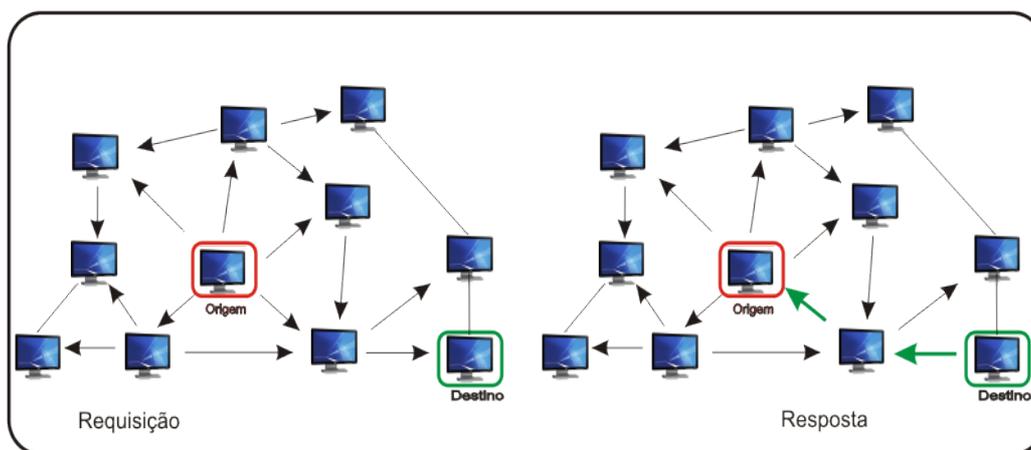


Figura 2 Exemplo de requisição usando a técnica de flood.

A técnica mais utilizada na implementação de arquiteturas puras, o qual gerou um avanço significativo na área, é o *Distributed Hash Tables* (DHT)[20]. Utilizadas nas ferramentas pStore [1], Pastiche [2], Oceanstore [3], PeerStore[25] e BitTorrent[15]. As DHTs pertencem à classe de sistemas distribuídos descentralizados e oferecem recursos de localização similar às *hash tables* (chave, valor). Um par de chaves e valor é armazenado na DHT e qualquer participante da rede pode acessar o valor desejado apenas informando a chave associada. As primeiras quatro especificações de DHTs (Pastry [16], Chord [17], CAN[18] e Tapestry [19]) surgiram quase simultaneamente no ano 2001, depois disso, sua utilização se popularizou em aplicações destinadas ao compartilhamento de arquivos na internet. Um estudo mais aprofundado sobre as diferentes implementações de DHT pode ser encontradas em outros trabalhos [20].

As DHTs possuem como principais características:

- Descentralização: os próprios nós criam e mantêm o sistema, sem a necessidade de um servidor;
- Escalabilidade: o sistema suporta a participação de um crescente número de nós simultaneamente;
- Tolerância a erros: o sistema deve ser confiável, mesmo com nós entrando e saindo continuamente da rede.

Para alcançar os objetivos supracitados, as redes DHTs utilizam a técnica de que um nó na rede deve estar em contato direto com apenas uma fração de todos os nós participantes. Isso reduz o custo de manutenção quando um nó entra ou sai do sistema.

Para armazenar um arquivo numa DHT, primeiro se calcula uma chave (geralmente o código *hash* SHA-1 [21] do seu nome ou do seu conteúdo), em seguida esse arquivo é enviado para a rede até ser encontrado o conjunto de nós responsáveis por seu armazenado. Para recuperá-lo, uma mensagem é enviada informando a chave do arquivo desejado. Essa mensagem por sua vez é encaminhada até um nó que possui o conteúdo desejado, que é enviado como resposta. A seguir é descrito uma das implementações de DHT mais utilizadas: o Chord [17].

1.2.3 Arquitetura Chord

A implementação de DHT utilizando Chord se destaca pela sua simplicidade em oferecer uma única operação em que dada uma determinada chave, ela será mapeada para um nó na rede. Segundo Stoica [17], sua arquitetura foi projetada para se adaptar facilmente à entrada e saída de novos *peers* na rede.

Embora seja uma implementação que possui apenas uma tarefa (associar chaves a nós da rede), o Chord possibilita ao desenvolvimento de aplicações P2P uma série de benefícios:

- Balanceamento de carga: as chaves são distribuídas igualmente entre os nós da rede;
- Descentralização: nenhum nó é considerado mais importante que outro;
- Escalabilidade: o uso do Chord é viável mesmo com uma grande quantidade de nós;
- Disponibilidade: ajuste de sistema automático à entrada e saída de novos nós, fazendo que um nó sempre esteja visível na rede;
- Flexibilidade: não é necessário seguir nenhuma regra para o nome das chaves.

Na Figura 3 [17] é ilustrada a localização e armazenamento de dados que podem ser facilmente desenvolvidos acima da camada de Chord, associando a chave à informação no nó a que ela faz parte no sistema.

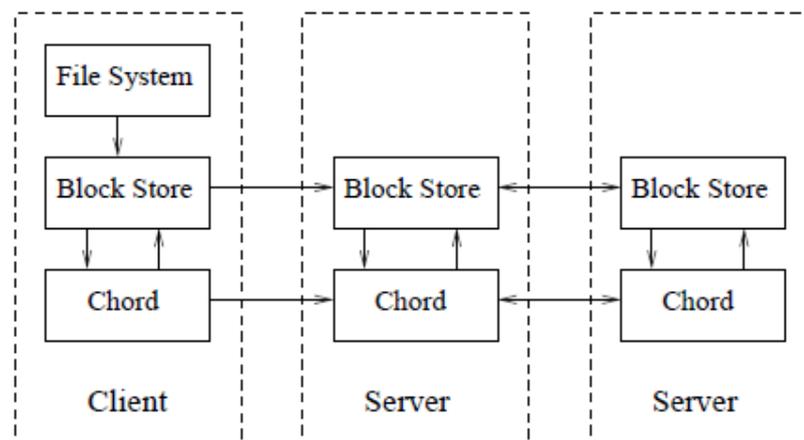


Figura 3 Sugestão de arquitetura de um sistema de armazenamento utilizando Chord. [17]

Para atribuir chaves aos nós, o Chord usa uma variação de *consistent hashing* [22] que cuida do balanceamento de carga, uma vez que cada nó tende a receber naturalmente um mesmo número de chaves. Em trabalhos anteriores ao Chord usando *consistent hashing*, se assumia que cada nó possuísse conhecimento de todos os outros, diferentemente, no Chord cada nó precisa ter conhecimento de apenas uma fração dos outros nós na rede.

Supondo uma rede de n nós, um *peer* mantém informações sobre $O(\log n)$ nós, e para encontrar um determinado nó na rede basta que ele possua apenas uma referência válida.

1.2.4 Arquiteturas híbridas

Em alguns sistemas P2P é necessária a identificação dos *peers* conectados na rede. Para tal, sistemas como o OurBackup[1], que fazem uso de redes sociais para *backup*, utilizam em sua arquitetura um servidor responsável pela autenticação dos usuários, manutenção da rede e dos metadados onde as cópias estão armazenadas. Pode-se ressaltar que a utilização de servidores não é obrigatória para a localização dos *peers* e dos metadados, podendo essa ser feita utilizando as DHT mencionadas anteriormente.

Nesses sistemas, o papel do servidor está em oferecer uma interface aos *peers* da rede com diversas operações tais como: autenticação do usuário; manipulação dos dados armazenados por outros *peers* (adicionar, remover, excluir, atualizar); manipulação dos usuários cadastrados no sistema; localização dos *peers* e relacionamento entre eles; dentre outras tarefas que venham a atender os requisitos do sistema.

Essa centralização de informações pode trazer prejuízos de escalonamento no sentido de que sempre vai existir uma exigência maior do servidor à medida que se aumenta o número de requisições, usuários, metadados ou quaisquer outras informações delegadas ao servidor. Porém, sistemas como o eDonkey [23] se mostraram bastante eficientes quanto ao gerenciamento centralizado de informações dessa natureza. Se necessário esse escalonamento também pode ser resolvido com a utilização de *clusters* ou *grids* no lado do servidor, fazendo-se mais importante a disponibilização da interface de comunicação com a máquina cliente.

1.2.5 Segurança e compartilhamento de pedaços de arquivos

Em busca de garantir a segurança e a possibilidade de que diferentes usuários possam compartilhar os mesmos blocos (pedaços) de arquivos diferentes, o identificador de cada bloco é calculado utilizando-se a seguinte fórmula: $ID = h(h(c))$; onde c é o conteúdo do bloco e h é uma operação de criptografia *hash*. Possibilitando, assim, a identificação e compartilhamento, de forma segura, dos blocos semelhantes, mesmo que seu conteúdo não seja conhecido.

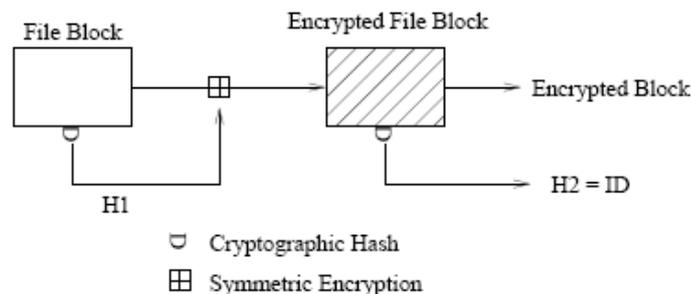


Figura 4 Criptografia dos blocos e geração do identificador. [25]

Na Figura acima se pode ver uma ilustração do modelo descrito acima, na proposição do sistema PeerStore [25]. Nessa abordagem o valor *hash* calculado a partir do conteúdo do bloco é usado como chave para criptografia do mesmo. Por essa razão, esse valor não é utilizado diretamente como identificador, por isso um novo valor *hash* é calculado. Dessa forma, apenas os *peers* que geraram o bloco podem ter acesso ao seu conteúdo, uma vez que geraram o mesmo identificador para blocos semelhantes.

1.2.6 Verificação da consistência dos blocos

De forma a verificar a consistência dos blocos que foram armazenados remotamente, um *peer* deve desafiar periodicamente os seus parceiros para saber se ainda estão guardando as cópias dos blocos a eles confiada. Contudo, solicitar que todos os blocos sejam recuperados apenas para teste não se mostra viável, uma vez que o consumo de banda de rede para tal fim seria muito alto. Assim como o Samsara [8], o sistema PeerStore [25] propõe a forma descrita na figura abaixo para a solução desse problema.

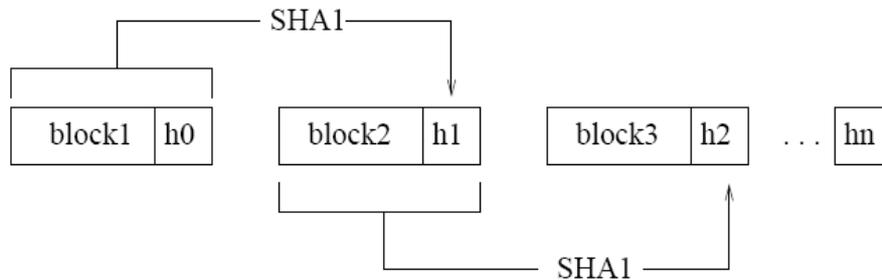


Figura 5 Verificar consistência dos dados armazenados apenas enviando h_0 e h_n . [25]

Nessa técnica, o *peer* que deseja verificar a consistência dos dados envia um valor único h_0 , assim como uma lista dos n blocos em questão. Para responder, o *peer* desafiado deve concatenar o valor h_0 ao conteúdo do primeiro bloco e calcular o código *hash* desse novo valor, gerando assim h_1 . O novo valor h_1 é então concatenado ao conteúdo do segundo bloco onde um novo valor *hash* h_2 será calculado, e assim sucessivamente até que o valor de todos os blocos seja verificado gerando como resultado final h_n . Esse valor final é então enviado como resposta provando, assim, a presença de todos os blocos.

1.2.7 Disponibilidade

Segundo a norma ISO/IEC 9126 [3], a confiança pode ser definida como sendo a “capacidade do produto de software de manter um nível de desempenho especificado, quando usado em condições especificadas”.

Avaliar se um sistema é confiável ou não envolve analisar dados quantitativos e qualitativos. Com base nesse contexto, Sommerville [1] classifica essa avaliação em quatro pilas. São elas:

- **Disponibilidade:** é a probabilidade de um sistema, em determinado instante, ser operacional e capaz de fornecer os serviços requeridos;
- **Confiabilidade:** é a probabilidade de operação livre de falhas durante um tempo especificado, em um dado ambiente para um propósito específico;
- **Segurança:** é um atributo que reflete a capacidade do sistema de operar, normal e anormalmente, sem ameaçar as pessoas ou ambiente;
- **Proteção:** é a avaliação do ponto em que o sistema protege a si mesmo de ataques externos, que pode ser acidentais ou deliberados;

Disponibilidade e confiabilidade são em essência probabilidades e, portanto são expressos em valores numéricos. Já segurança e proteção são feitos com base em evidências na organização dos sistemas e normalmente não são expressos em valores numéricos. Frequentemente são medidos através de níveis, e a ordem de grandeza destes denomina se um sistema é mais ou menos seguro/protegido do que outro.

Oliveira [1] concorda com Sommerville [1], quando define disponibilidade em sistemas de *backup* P2P como sendo a probabilidade de recuperação de uma informação num determinado instante t . Logo, quando se fala sobre armazenamento de dados em sistemas P2P, sua natureza distribuída contribui para uma maior disponibilidade, uma vez que a distribuição geográfica intrínseca dos pontos de rede reduz as chances de falha simultânea por acidentes catastróficos. No entanto, computadores em uma rede P2P estão disponíveis apenas uma fração de tempo, ou estão sujeitos as mais diversas falhas, o que pode trazer prejuízos a disponibilidade dos dados armazenados. Para reduzir esse risco e aumentar a disponibilidade do sistema, faz-se necessário o uso de redundância dos dados, seja por replicação ou usando técnicas mais elaboradas como o *erasure codes* [2].

1.2.8 Replicação dos dados

A replicação é a mais tradicional forma de garantir disponibilidade, onde k cópias idênticas de um dado é copiada em diferentes instâncias dos pontos de rede. O número k deve ser definido de forma a aumentar a probabilidade de recuperação de alguma informação dada a probabilidade de um *peer* que a possua estar disponível no instante desejado. O aumento probabilístico da disponibilidade é obtido somando-se as probabilidades de apenas um nó estar disponível, dois nós estarem disponíveis, três nós estarem disponíveis, até k . [1] Ou, esse somatório pode ser das probabilidades de as cópias terem falhado no lugar de as cópias estarem disponíveis. Sendo assim, essa disponibilidade pode ser descrita segundo a seguinte equação, onde f é a falha e k a quantidade de cópias:

$$C = \sum_{i=1}^k \binom{k}{i} (1-f)^i (f)^{k-i}$$

Equação 1 Cálculo da probabilidade de que uma informação esteja disponível da rede.

Na tabela a seguir são mostrados os resultados obtidos na aplicação da formula.

Tabela 1 Resultados obtidos na aplicação da fórmula para 20% (esquerda) e 30% (direita) de falha.

f	K	C	f	k	C
20%	4	0.998400	30%	6	0.999271
20%	5	0.999680	30%	7	0.999781
20%	6	0.999936	30%	8	0.999934
20%	7	0.999987	30%	9	0.999980
20%	8	0.999998	30%	10	0.999994

Note que para uma taxa de falha de 20% se podem obter cinco noves de precisão efetuando oito cópias da informação, e para uma taxa de falha de 30% podemos obter um resultado bastante similar aumentando o número de cópias em duas.

1.3 Sistemas em Grids

1.3.1 Conceitualização

A literatura sobre *Grid Computing* abrange diversas definições sobre este conceito. Porém, não há uma concordância geral entre os termos utilizados. Para Aidouni *et al.* [21] um *grid* computacional é definido como uma infra-estrutura que envolve o uso colaborativo e integração de computadores, redes, banco de dados e instrumentos científicos pertencentes e gerenciados por múltiplos domínios administrativos visando compartilhar recursos computacionais além dos limites institucionais, a fim de prover uma plataforma poderosa e distribuída de altíssima escala, com um custo muito inferior quando comparada a um supercomputador paralelo.

Por outro lado, para Ian Foster [22], Grid é uma infra-estrutura de software e hardware para integrar recursos computacionais, dados e pessoas dispersas geograficamente formando um ambiente colaborativo distribuído.

Já Krauter [23], define *grid* como um sistema de rede que pode escalar ambientes do tamanho da internet com máquinas geograficamente distribuídas através de múltiplas organizações e domínios administrativos.

1.3.2 Abordagem Histórica

Na década de 80, as primeiras manifestações de computação em grade consistiam em aplicativos que utilizam o paradigma de computação paralela para a promoção de mecanismos de comunicação entre processadores, proporcionando um grande potencial para realizar compartilhamento de memória [24].

A fim de possibilitar novos achados científicos, a comunidade científica se viu com o desafio de trabalhar em pesquisas multidisciplinares nas quais envolviam grandes quantidades de dados a serem processados, disponibilizados e apresentados [24]. A chave principal desse desafio estava em como promover uma infra-estrutura computacional em rede de larga escala e de baixo custo [24][25].

No início da década de 90, impulsionados pelo uso de rede de computadores e computação de larga escala, surgiram uma gama de projetos que faziam uso de computação distribuída. Em 1995, durante uma conferência promovida pela IEEE e ACM sobre computação paralela, foram utilizados 11 redes de computadores para demonstrar a criação de um supercomputador distribuído geograficamente [26]. O projeto foi conduzido por Ian Foster e recebeu o nome de I-Way (*Information Wide Area Year*), ficando conhecido como o primeiro modelo de implementação de grades computacionais [25].

O sucesso do projeto incentivou diretamente a criação de mais de 70 aplicações, como também sua continuidade, que foi patrocinada por diversas instituições governamentais norte-americanas [26]. Consequentemente, no ano seguinte, a DARPA (*Defense Advanced Research Projects Agency*), desenvolveu e financiou um projeto de pesquisa para a construção de uma ferramenta de computação distribuída em larga escala conhecida como GLOBUS, conduzido também por Ian Foster e pelo pesquisador Carl Kesselman [25][24].

Durante a Conferência de Supercomputação(SC98), um fórum foi estabelecido para estudar, discutir e implementar padrões e melhores práticas no uso de grades computacionais. Em 2000, o eGrid (*European Grid Fórum*) e o *AsianPacific Grid Fórum* se uniram e fundaram o *Global Grid Fórum*. Anos mais tarde, ocorreu outra fusão entre a *Global Grid Fórum* e a *Enterprise Grid Alliance* (EGA), que resultou na fundação do *Open Grid Fórum* (OGF) [27].

O *Open Grid Fórum* é uma comunidade de usuários e desenvolvedores que lideram estudos de padronização para grades computacionais. A comunidade consiste em milhares de pessoas especializadas na indústria e na pesquisa, representando mais de 400 organizações em mais de 50 países ao redor do mundo.

1.3.3 Visão Geral sobre Grids

O termo *grid* surgiu nos Estados Unidos como referência a rede de distribuição de energia elétrica (*Power grids*). Semelhante a essas redes, existe uma transparência de como o recurso utilizado está sendo de fato produzido ou armazenado. Como não é possível armazenar a energia elétrica distribuída pela rede, ela deverá ser consumida ou será perdida.

Analogamente a essas redes, os ciclos de processamento de CPU também não podem ser armazenados. Consequentemente, em períodos de ociosidade do processador, o processamento poderia ser efetuado sem o usuário final saber onde exatamente a computação está sendo executada.

A infra-estrutura de ambas as redes são heterogêneas e são formadas por diferentes componentes: usinas de produção de energia elétrica, linhas de transmissão, transformadores e, do outro lado, computadores, servidores, conexões de rede, dispositivos de armazenamento. Adicionalmente, tanto redes elétricas como *grids* computacionais são pervasivas, ou seja, o usuário pode ter acesso ao recurso desejado em qualquer parte da rede onde exista um ponto de entrada, como ilustrado na figura 6 [28].



Figura 6 Visão Geral de Grid Computacional

Este modelo distribuído de arquitetura computacional traz diversos benefícios além do compartilhamento dos ciclos de CPU e espaço de armazenamento. Ele também provê a capacidade de processar paralelamente aplicações subdivididas em partes menores, que podem ser distribuídas em milhares de computadores ao mesmo tempo visando aumentar o desempenho dessas aplicações. Essa abordagem se mostra muito interessante do ponto de vista da confiabilidade do sistema, visto que a ocorrência de falha em pontos específicos do *grid* não compromete o funcionamento total da rede [29].

Para controlar o acesso a todos os recursos de um *grid* é necessária uma camada de *software* que fica localizada entre a interface com o usuário e o sistema operacional. Essa camada de software é denominada *middleware*. Ele é geralmente constituído por módulos com APIs de alto nível, que proporcionam a sua integração com aplicações desenvolvidas em diversas linguagens de programação, e interfaces de baixo nível que permitem a sua independência do restante da rede [29], como está ilustrado na figura 7.

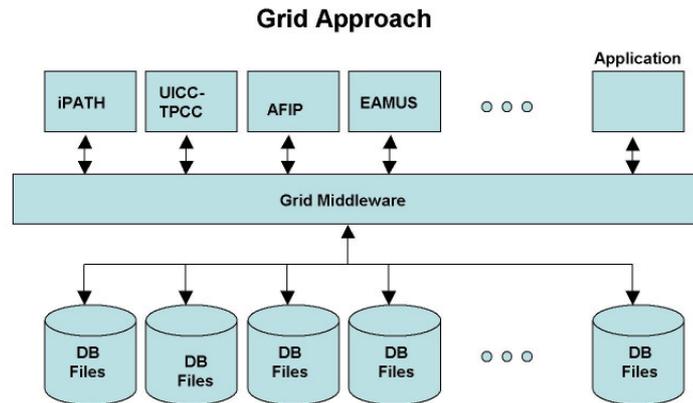


Figura 7 Camada de Software Middleware

Segundo Ian Foster [30], existem três características básicas que compõem *Grids* Computacionais:

1. Um *Grid* não possui um controle centralizado, onde as questões de segurança, políticas de acesso e controle do uso de recursos são consideradas dentro dos diferentes domínios que constituem o *Grid*;
2. O *Grid* é formado por protocolos abertos e padronizados para proporcionar alto grau de interoperabilidade no ambiente;
3. O *Grid* deve fornecer QoS (Qualidade de Serviço) com o propósito de atender diferentes requisitos de tempo de resposta, vazão, disponibilidade, segurança e alocação de diversos recursos dentro de um sistema heterogêneo e complexo.

A ideia de compartilhamento de recursos não é nova, pois a própria *web* pode ser considerada como um serviço de compartilhamento de informações sobre a camada da internet. Desde as iniciativas pioneiras em grades computacionais até os projetos mais recentes, todos procuram explorar a ociosidade das máquinas para realizar análises e cálculos que seriam feitos exclusivamente em um único servidor central.

1.3.4 Organizações virtuais

Grupos que estão dispersos geograficamente e que estão dispostos a doar tempo de processamento ocioso para a resolução de problemas podem ser agrupados no que chamamos de organizações virtuais (*Virtual Organization - VO*). Uma VO pode ser comparada a um *cluster* ou um agrupamento de computadores que estão sob o mesmo domínio administrativo, no qual são determinadas regras de compartilhamento de recursos. É importante ressaltar que a união de várias VO's através de redes de computadores (LAN – *Local Area Network* ou MAN – *Metropolitan Area Network*) é o que realmente forma uma *grid* na sua forma mais conceitual. Na figura 8 abaixo é ilustrado a formação de um *grid* computacional com a disposição de várias VO's ao redor do mundo.

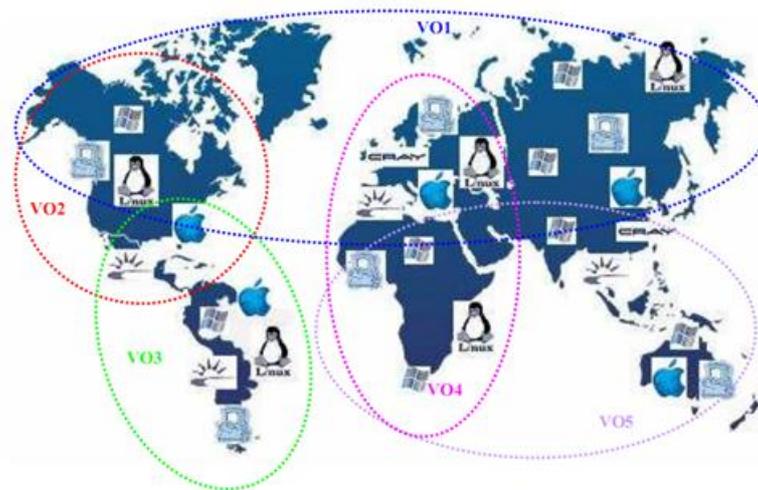


Figura 8 Organizações Virtuais formando uma grid

1.3.5 Componentes de uma grade computacional

Durante a evolução do estudo e implementações de grades computacionais, diversas arquiteturas foram propostas sem seguir nenhuma padronização comum (comunicação, dados, plataformas e protocolos), gerando um impacto negativo na interoperabilização dos sistemas.

Com a melhoria e aperfeiçoamento dos protocolos e tecnologias aplicadas na Internet e nas próprias grades computacionais, se estabeleceu uma componentização mínima necessária para o estabelecimento de ambientes de grade, sendo formado por: interface do usuário, segurança, agente de busca de informações, escalonadores, gerenciamento de dados e gerenciamento de tarefas e recursos. Essas funcionalidades serão melhor descritas nas próximas seções.

1.3.6 Interface do Usuário

Um ambiente de *grid* computacional pode disponibilizar diversos tipos de interface com usuário. Dentre as mais utilizadas, é possível destacar: interpretador de comandos, aplicação cliente pré-definida e portais de serviços (*web services*).

A alocação, distribuição e execução de tarefas através de um interpretador de comandos é o mecanismo mais primitivo porém mais usado e preferido por usuários mais experientes. Através dele é possível inserir comandos ou até criar *scripts* que serão executados e interpretados por um interpretador comumente chamado de *shell*.

As aplicações cliente dedicadas a exibição da interface gráfica são aplicativos *desktop* que rodam na estação do usuário e possuem um canal de comunicação com a grade. Esse tipo de aplicação executa uma lista limitada de tarefas e comandos simples, requisitando ou executando-os na grade. Um exemplo desse tipo de operação seria a submissão de dados por meio de arquivos em lote (sistemas *batch*) ou utilizando XML.

Portais *web* são as interface mais utilizadas na atualidade devido a expansão da Internet em todo mundo. Um portal de grade pode disponibilizar uma série de serviços para uma determinada comunidade, necessitando apenas que os usuários cumpram os requisitos mínimos de acesso. O mais interessante desse tipo de interface é que os administradores podem incluir ou remover serviços dinamicamente sem interferência nas estações dos usuários.

1.3.7 Segurança

Para entender os problemas de segurança envolvidos em sistemas distribuídos que utilizam os conceitos de *grids*, é conveniente apresentar um cenário genérico que ilustre um ambiente de computação distribuída e seus recursos envolvidos em instituições distintas. Na figura 9 é apresentado este cenário.

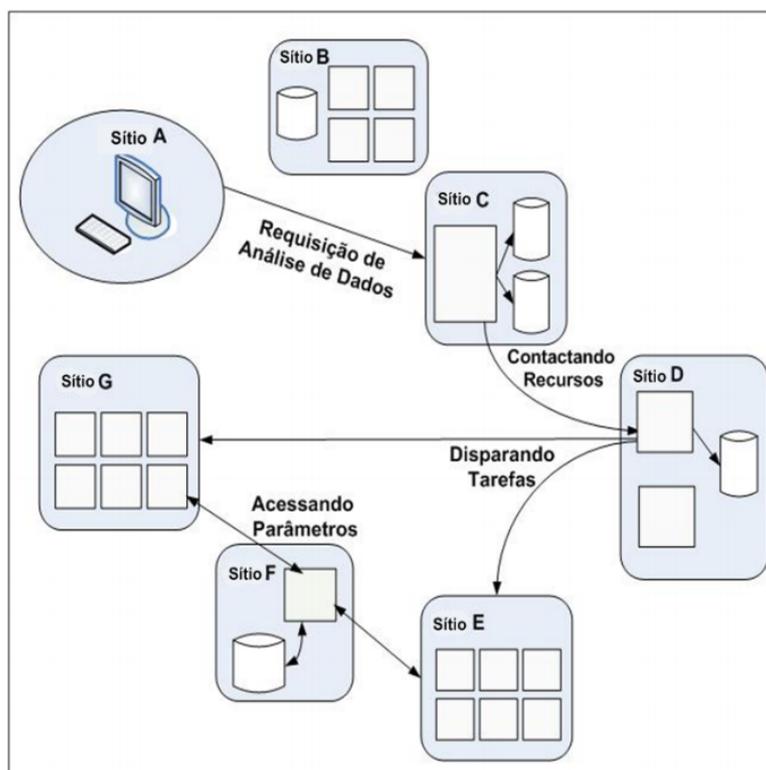


Figura 9 Arquitetura de segurança para grids

Dentro desse cenário é possível observar que existem sete sítios (A-G) que dispõem de recursos e necessidades distintas. Os processos de comunicação entre esse sítios (representados pelas setas de interligação) são realizados da seguinte maneira: A solicita um procedimento de análise de dados para C; C conecta-se com D para adquirir informações sobre recursos; D solicita e distribui tarefas para G e E; G e E consultam parâmetros em F.

Foster descreve os seguintes pontos a serem considerados num cenário clássico de funcionamento de uma *grid* computacional:

1. Os usuários das organizações multi-institucionais formam um grupo extenso e dinâmico. Os participantes desses grupos, constituintes da organização virtual, podem ser alterados, incluídos ou excluídos dependendo da necessidade e da quantidade de recursos disponíveis pela instituição.
2. A disponibilidade de recursos é vasta porém dinâmica.
3. O processamento dos recursos computacionais apresentam características distribuídas.
4. Os processos podem criar e destruir conexões (*unicast* ou *multicast*) dinamicamente em tempo de execução.

- Os recursos podem ser acessados por diferentes formas de autenticação, autorização e política devido a divergência do ambiente de segurança empregado pelas instituições participantes.

A política de proteção de uma infra-estrutura de grade visa prevenir ataques aos recursos disponibilizados pela rede, congestionamento de recursos e a inclusão de dados espúrios que impossibilitem a legitimidade da informação. Consequentemente, o ambiente deve contemplar políticas de autenticação, controle de acesso, integridade, confidencialidade e irretratabilidade.

1.3.8 Agentes de Busca de Informações

O agente de busca de informações, ou *Broker*, é o componente responsável por manter um banco de informações sobre a disponibilidade de recursos e serviços que fazem parte da rede, além de informar aos usuários conectados. Também compete ao *Broker* a atividade de controlar a informação do estado (disponível, trabalhando, ocupado, indisponível ou falho) e da localização de recursos através de descrições publicadas de nome e de diretório (DNS e LDAP).

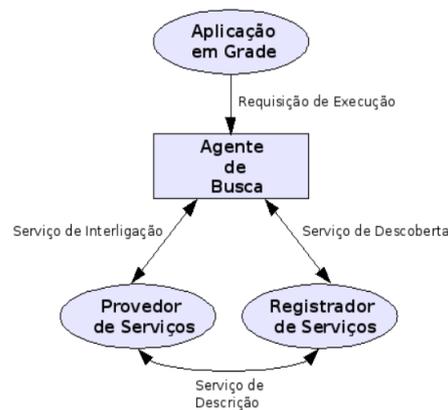


Figura 10 Agentes de busca

A figura 10 apresenta como o *Broker* realiza a busca por informações de recursos disponíveis na rede para execução de uma tarefa específica. Quando há uma necessidade por parte da aplicação de realizar essa busca, uma requisição é disparada para o agente de busca que, por sua vez, consulta o provedor e registrador de serviços em busca da disponibilidade do recurso no ambiente.

1.3.9 Escalonadores

Escalonadores são componentes responsáveis pela coordenação da execução dos processos computacionais. Esses componentes são aplicados exaustivamente em sistemas operacionais multiprogramados para o escalonamento de processos e execução de aplicações paralelas.

Um exemplo de atividade que um escalonador realiza é a alocação de tarefas computacionais para um conjunto pré-definido de unidades de processamento. Esse método é normalmente empregado em *clusters*, onde os processadores são, na maioria dos casos, homogêneos.

Os escalonadores desenvolvidos ou adaptados para o uso em grades computacionais são conhecidos como meta-escalonadores (*metascheduler*). Os meta-escalonadores são compostos por ferramentas para designar prioridade de execução, re-submissão de tarefas e escalonamento multinível. Para a execução desse tipo de tarefa, os componentes da rede recebem informações sobre a capacidade do processamento disponível e informações sobre o estado dos recursos.

Os escalonadores para grades computacionais são compreendidos como sendo o escalonador global dos escalonadores locais e equipamentos que compõem a infra-estrutura. Pelo fato da interação e submissão de tarefas na grade computacional, um meta-escalonador precisa ter compatibilidade com ferramentas de segurança, sistemas de informação e ambiente de execução da grade. A figura 11 apresenta a visão esquemática de uma arquitetura que é aplicado metaescalonamento.

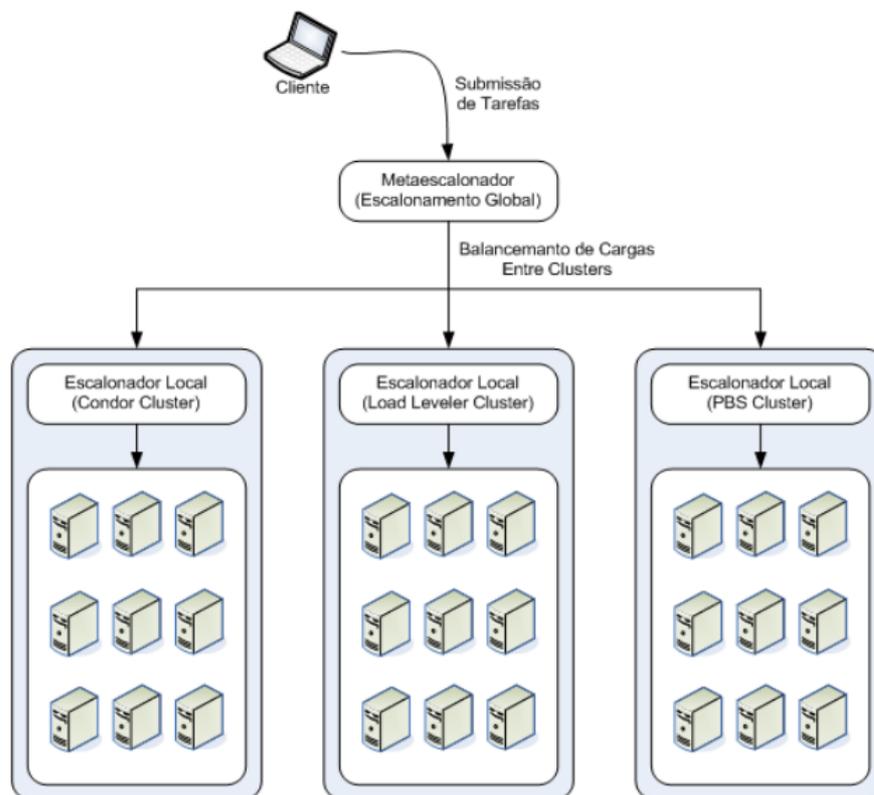


Figura 11 Visão geral de uma arquitetura com metaescalonador

1.3.10 Gerenciamento de Dados

O gerenciamento de dados ainda é um desafio para as aplicações em grades computacionais, pois o modelo de gerenciamento de dados é dependente da taxonomia ou finalidade do ambiente. Dentro de uma grade computacional podem ser encontrados diferentes tipos e formatos de dados que devem ser operados de forma comum. Alguns exemplos dessa variabilidade são: arquivos convencionais, base de dados relacionais, base de dados XML, objeto de dados, dados virtualizados, setores de dados e metadados.

A gerência e a movimentação de arquivos ou diretórios fica a cargo dos protocolos convencionais utilizados na Internet, que passaram por uma adaptação para se adequarem ao ambiente de grades computacionais, como é o caso do GridFTP. Nos demais tipos de dados se torna indispensável uma camada de software adicional (*middleware*) que possibilite a manipulação de tais informações com mais transparência. A função básica dessa cama de

software é compreender o provedor de informações (i.e. SGBD) e transformar essas informações num padrão comum, tal como os documentos em XML.

Na figura 12 é mostrado um modelo de acesso à base de dados relacionais em ambientes de grade computacionais.

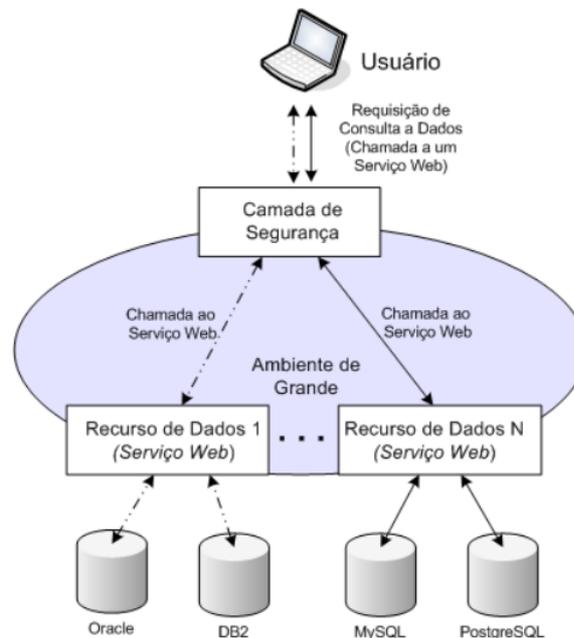


Figura 12 Gerenciador dos dados

1.3 Segurança em computação nas nuvens

Segurança em *cloud computing* não difere muito da segurança tradicional. No entanto, se deve ter mais atenção no software do que na infra-estrutura, visto que esta última, no melhor dos casos, quando se trata de um modelo de Infra-Estutura como Serviço (IaaS), estará sobre controle do desenvolvedor, com exceção do *hypervisor*.

Sendo assim, se faz necessário a proposta de uma metodologia que vise a garantia da segurança de um software que não cause grandes acréscimos de tempo e custo ao ciclo de desenvolvimento. Esse é um tema relevante e atual visto que a maioria das falhas reportadas nos sistemas estão nas aplicações e não mais na infra-estrutura, transformando a computação nas nuvens o local onde esse problema é potencializado.

No entanto, a existência de processos consolidados para garantir a segurança da infraestrutura não garante a segurança como um todo, visto que uma aplicação insegura em uma rede segura é inseguro da mesma forma.

Um sistema é considerado seguro quando a segurança está presente tanto na infraestrutura, quanto na aplicação. Porém, uma falha em qualquer um dos dois componentes e/ou em seus processos pode comprometer toda a segurança.

Considerando a evolução da Engenharia de Software, temos que qualidade de um software não é apenas garantida através da execução de um processo, mas sim a execução de testes e validação [30] no intuito de garantir que o produto entregue reflita a visão do cliente, ao menos em termos de especificações. Sendo assim, vemos práticas como TDD (*Test Driven Development*) e outras recomendações, em conjunto de processos e metodologias, sendo incorporados ao processo de desenvolvimento de software.

A Figura 13 apresenta o ciclo de segurança para um software em *cloud computing*:

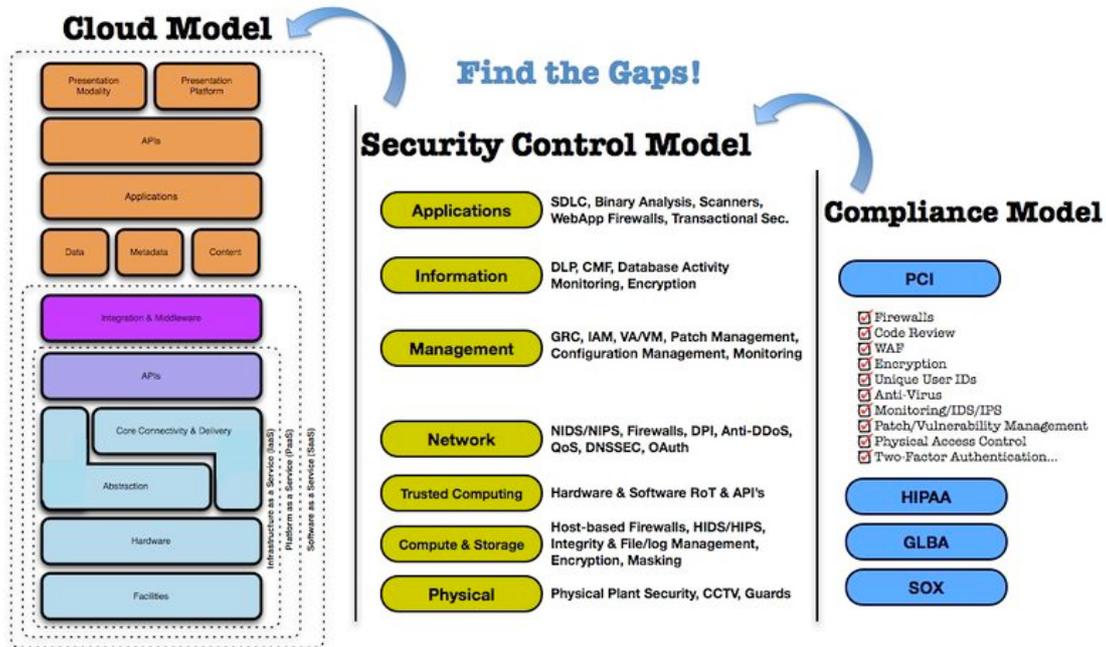


Figura 13 Segurança em Cloud Computing (Fonte: Cloud Security Alliance)

1.3.11 Processo para a garantia da qualidade

De posse das considerações feitas, a proposta de um processo de garantia da qualidade focado em segurança e que seja ágil e focado no reuso de software deve transpassar as atividades de:

a) **Elaboração dos requisitos:** a elaboração dos requisitos de segurança de softwares demanda profissionais mais experientes, pois exige conhecimento sobre segurança tanto para a utilização de estratégias de defesa como a execução de testes de validação da segurança. Como descrito, o conjunto de ataques mais frequentes à uma aplicação é conhecido, logo, pode-se basear neles para escrita de requisitos bem elaborados que visem a defesa das aplicações. Como os ataques seguem um padrão, é possível definir requisitos reutilizáveis, fazendo com que os mesmos venham de uma base de conhecimento sobre tecnologias, padrões de projeto, arquiteturas e ataques.

b) **Teste e validação:** a garantia da qualidade é dada através da execução de testes no sistema. A elaboração e automação dos testes de penetração da aplicação baseando-se nos ataques que a aplicação irá sofrer, se mostra uma prática válida para garantir a segurança. Adicionalmente, o reuso dos requisitos e dos testes é facilitado, bem como permite os desenvolvedores garantirem que o software estará protegido sobre o ponto de vista dos ataques que ele está sujeito. Não se pode garantir 100% de segurança em uma aplicação, mas com esta abordagem se pode garantir aos clientes que sobre um conjunto conhecido de testes a aplicação está segura.

c) **Padrões de projeto:** em relação aos requisitos de segurança sabe-se que os mesmos podem variar com a evolução do projeto, principalmente com a adoção de metodologias ágeis. O desenvolvimento de software seguro deve considerar que este

fato irá acontecer. A solução para que a segurança acompanhe essas alterações está na adoção de padrões de projeto que permitam a implementação tardia dos requisitos.

Nas seções a seguir será discutido, detalhadamente, cada uma dos itens acima que permitem a elaboração softwares seguros de forma ágil. A figura 4 abaixo apresenta como este processo pode ser aplicado para o desenvolvimento de software seguro.

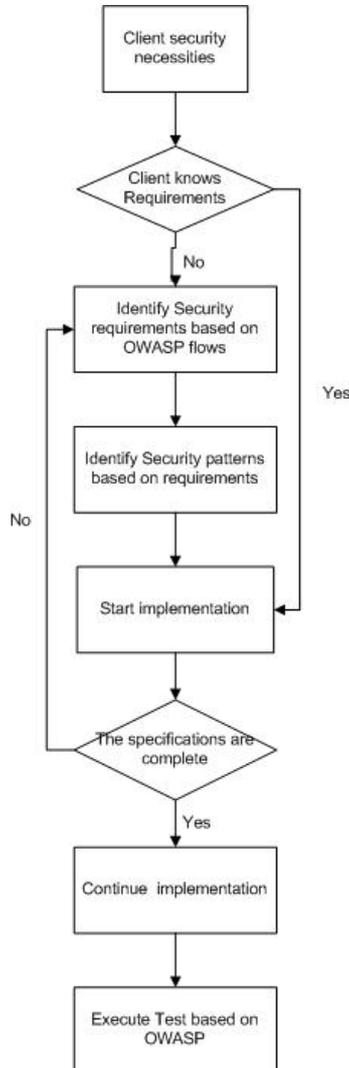


Figura 11 Processo para desenvolvimento seguro e ágil

1.3.12 Requisitos: Escrevendo requisitos de segurança

A solução proposta segue uma sequência de passos que se inicia na criação de um guia de segurança, passando para um *template* com a finalidade de facilitar a escrita dos requisitos de segurança, e como mapear, escrever e automatizar casos de testes de segurança, fechando o elo para a proposta de um processo para a garantia da segurança em projetos de software.

A figura abaixo mostra a solução proposta nesse trabalho para esta atividade, onde cada etapa mostrada será melhor descrita nas seções seguintes:



Figura 15 Solução proposta

A escolha da elaboração de um guia que dê suporte à escrita de requisitos ao invés de ir diretamente à escrita de requisitos através dos padrões para a escrita do mesmo. O guia é escrito baseado nos conceitos básicos de requisitos para web levando em consideração as necessidades de toda a equipe. Não só a análise de requisitos, mas qual a solução proposta para uma arquitetura mais segura, quais as opções de tecnologias existentes e quais são as vulnerabilidades associadas a cada requisito, sendo assim um guia prático para utilização de toda a equipe. O guia ajuda na divulgação e compartilhamento do conhecimento para toda equipe podendo ser utilizado para um passo inicial da gestão do conhecimento.

1.4.1 Guia

A criação de um guia de segurança para aplicações *web* se propõe não só a ajudar os engenheiros de requisitos a entender e detalhar as necessidades de segurança de forma mais profunda e clara, mas também para servir como referência para as outras fases do desenvolvimento do sistema. O guia foi formulado baseado nos padrões de requisitos proposto por [44] e no *template* especificação de documentação dos casos de usos desviados [31], bem como nas listas de falhas mais comuns e como testá-las em uma aplicação *web* proposto pelo OWASP [32].

O guia deve ser desenvolvido baseado no reuso das informações para diferentes tipos de sistemas *web*. Cada ponto do guia foi pensando para dar apoio a uma ou várias fases do desenvolvimento específico do ciclo de vida do desenvolvimento do sistema. Para isso foram formuladas algumas perguntas divididas por disciplinas, como mostra a tabela abaixo:

Requisitos

Quais são os requisitos comuns de segurança para aplicações web?

Aqui será listada uma serie de requisitos de segurança comuns às aplicações web e suas variações, de forma a ajudar o analista de sistema entender quais são as necessidades de segurança do cliente e como atingi-las.

Testes

Quais as vulnerabilidades encontradas em cada requisito?

Cada requisito ou objetivo de segurança está associado a várias vulnerabilidades e riscos que precisam ser mitigados, para isso primeiro precisamos entender quais são estas vulnerabilidades e qual o impacto da mesma no sistema.

Desenvolvimento

Quais são os métodos de soluções disponíveis?

Quais os padrões e soluções encontradas hoje no mercado e na academia para atender os requisitos levantados, e como esses se comportam e quais seus pré-requisitos.

Qual a ação para mitigar?

Agora que sabemos exatamente quais os possíveis ataques que podem ocorrer em nosso sistema, vamos analisar o risco e as possíveis formas de mitigar estes.

Qual a linguagem a ser utilizada?

Toda linguagem tem suas particularidades, desvantagens e vantagens, é importante saber quais são elas do ponto de vista de segurança das aplicações web e como podemos tratá-las e evitá-las.

Arquitetura e Design

Impactos da escolha arquitetural, frameworks e componentes?

Da mesma forma que a linguagem tem suas vantagens e desvantagens em relação à segurança, as decisões que tomamos na fase de análise e projeto, impactam de maneira significativa na segurança da nossa aplicação. Hoje existem padrões arquiteturais específicos para segurança [34]

O guia pode ser escrito e atualizado com a ajuda de toda a equipe, ou seja, a idéia é que cada perfil fique responsável pela atualização de cada ponto do guia:

- Analista de requisitos – ajudar na escrita dos requisitos de forma mensurável e atuar como facilitador da equipe;
- Engenheiro de Testes – levantamento das vulnerabilidades e como testar cada uma destas;
- Arquiteto – propor as soluções e padrões de arquitetura para as vulnerabilidades e problemas encontrados;
- Desenvolvedores – propor soluções de tecnologias e linguagens que podem ser utilizadas para mitigar os riscos levantados pela análise de impacto das vulnerabilidades.

Com o desenvolvimento do guia têm-se uma base de conhecimento para levantamento dos requisitos, além de ajudar a definir o que deve ser implementado no sistema e quais cuidados devem ter na hora de arquitetar e desenvolver o sistema baseando-se nas falhas e padrões de ataques associados a cada requisito, tendo ainda soluções de mitigação propostas para evitar ou radicalizar as vulnerabilidades e com isso reduzindo de forma significativa os riscos levantados.

Para a escrita do guia, foi desenvolvido um passo a passo a fim de atender e responder as perguntas acima levantadas. Desta forma, é usado uma forma estruturada de montar e atualizar o guia.

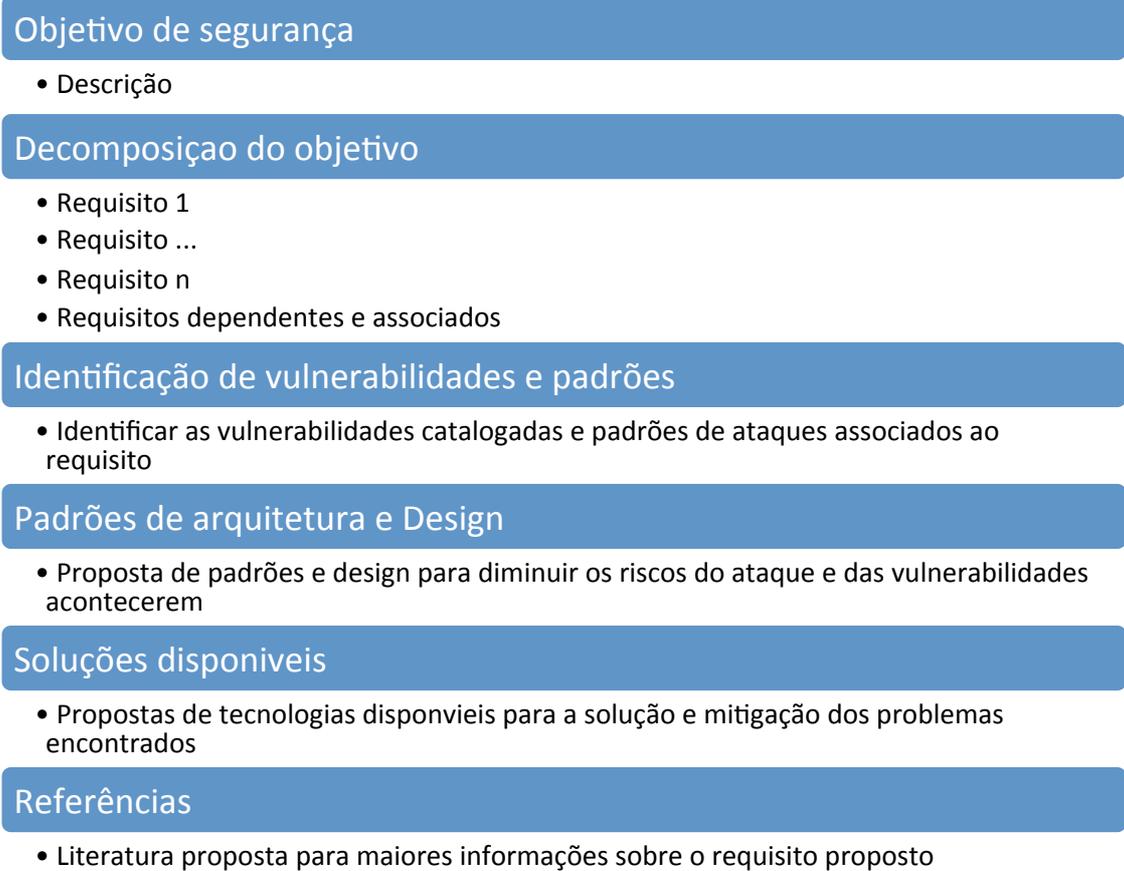


Figura 16 Estrutura do Guia de Segurança

Descrevendo a figura acima, encontram-se os seguintes pontos:

1. Objetivo – definir o objetivo do requisito de segurança. Quem vai ser decomposto em várias partes, ou seja, requisitos associados e requisitos dependentes que juntos vão suprir o objetivo de segurança;
2. Descrição – uma breve descrição do objetivo de forma geral e uma especificação do que significa para aplicações web;
3. Requisitos – decomposição do objetivo em requisitos funcionais e não funcionais, para que seja possível atingir o objetivo estabelecido no primeiro passo. Requisitos que estejam associados de forma indireta.
4. Vulnerabilidades – as vulnerabilidades relacionadas ao requisito e quais são os possíveis percas para o sistema, ocasionadas pela ameaça;
5. Padrões de arquitetura e design – padrões já catalogados na literatura;
6. Soluções disponíveis – tecnologias desenvolvidas e testadas;
7. Referências – literatura associada ao requisito.

1.4.2 Montando o documento de requisitos

Para montar o documento de requisitos vamos selecionar os objetivos de segurança e seus requisitos associados, que estão descritos no guia, e que são importantes para o sistema a ser desenvolvido. Neste caso montamos um *checklist* de acordo com o guia para que possa ser mais

fácil e usual de entender as necessidades do cliente. Neste *checklist* deve conter não só as informações contidas no guia. Algumas perguntas foram elaboradas de forma geral para ajudar no levantamento dos requisitos de segurança.

1. Existe uma política de segurança da empresa?
2. Existe um ambiente montado para a execução da aplicação?
3. Existe funcionalidades que são restritas?
4. Existe um controle de acesso centralizado?
5. Existe uma política de acesso aos dados?
6. Foi feita a classificação dos recursos e o de segurança atribuídas a ela?
7. Qual linguagem vai ser utilizada?

São perguntas que precisam ser respondidas. Caso não tenham sido pelo cliente, a equipe precisa definir quais as tecnologias serão utilizadas, como vai ser o acesso aos dados, entre outras coisas para poder desenvolver os requisitos de segurança do sistema.

Depois de respondidas essas perguntas, é preciso entender o domínio do sistema para selecionar, dentro desse domínio, quais são as necessidades de segurança do cliente. Para selecionar as necessidades é utilizado o guia de acordo com os objetivos de segurança. Por exemplo, se o sistema vai ter dados de acesso restrito, temos como objetivos a integridade dos dados, controle de acesso, e dentro deles os requisitos decompostos. O *checklist* é utilizado para selecionar os que são adequados e possíveis ao sistema a ser desenvolvido.

Tabela 2 Exemplo de *checklist* de requisitos de segurança

Objetivo: RFN001 Integridade dos dados

Requisito	Sim	NA	Comentários
RFN001_1 Transferência de dados			
RFN001_2 Classificar informações			
RFN001_3 Acessar informações			
RFN001_4 Armazenamento de dados			
RFN001_5 Validação dos dados			
RFN001_6 Encode de dados			

1.4.3 Calculando o risco

Para que seja possível a utilização de metodologias ágeis em um processo de desenvolvimento de software considerando segurança, é necessária uma lista de requisitos reutilizáveis. Caso seja necessário que os mesmos sejam considerados para implementação, que o gerente do projeto possa calcular o risco que esta nova funcionalidade que tenha, são necessários dois fatores:

- A probabilidade de um invasor explorar determinada falha depende das decisões de arquitetura do sistema, bem como de algumas questões relacionadas à implementação.
- O cálculo do impacto que uma determinada falha pode causar a um sistema e que pode ser previamente definida através do uso de metodologias.

1.4.3.1 Cálculo do impacto

Para calcular o impacto é utilizado o método disponibilizado pelo NIST: o CVSS – *Common Vulnerability Scoring System* [35]. Este consiste em três grupos: Básico, Temporal e Ambiental. Cada grupo produz um resultado numérico variando de 0,0 a 10,0, e um vetor, uma representação textual que reflete os valores utilizados para obter a pontuação. O grupo Base representa as qualidades intrínsecas de uma vulnerabilidade. O grupo temporal reflete as características de uma vulnerabilidade que mudam ao longo do tempo. O grupo Ambiental representa as características ambientais de uma vulnerabilidade que são únicas para qualquer usuário do ambiente. CVSS permite que gestores, provedores de boletim de vulnerabilidade, fornecedores de segurança, vendedores de aplicações e pesquisadores se beneficiem de uma linguagem comum de pontuação das vulnerabilidades de TI. Um exemplo de como calcular o impacto é apresentado abaixo:

Para calcular o impacto utilizamos a vulnerabilidade: V1 - Vulnerabilidade associada: Transporte de credenciais através de canal criptografado (OWASP-AT-001)

Fazendo o cálculo do grupo de métricas base:

Métricas de exploração:

1. Vetor de acesso: *Network* (rede)
2. Complexidade de acesso: *Medium* (Média)
3. Autenticação: *None* (nenhuma)

Métricas de Impacto:

1. Impacto de confiabilidade: *Partial* (parcial)
2. Impacto de integridade: *None* (nenhum)
3. Impacto de disponibilidade: *None* (nenhum)

Resultado obtido:

CVSS Base Score	7.8
Impact Subscore	7.8
Exploitability Subscore	8.6
CVSS Temporal Score	Undefined
CVSS Environmental Score	Undefined
Overall CVSS Score	7.8

Figura 17 Exemplo de cálculo de impacto do grupo base

Atualizando o impacto utilizando as métricas de ambiente:

Modificadores gerais:

1. Potencial danos colaterais: *Low-Medium* (Baixo moderado)
2. Alvo de distribuição: *Medium* (médio)

Sub pontuação do impacto dos modificadores:

1. Requisito de confiabilidade: *Medium* (médio)
2. Requisito de integridade: *Low* (baixa)
3. Requisito de disponibilidade: *Low* (baixa)

Resultado obtido:

CVSS Base Score	4.3
Impact Subscore	2.9
Exploitability Subscore	8.6
CVSS Temporal Score	Undefined
CVSS Environmental Score	4.5
Modified Impact Subscore	2.9
Overall CVSS Score	4.5

Figura 18 Exemplo de cálculo de impacto do grupo ambiental

Atualizando o impacto utilizando as métricas de tempo:

1. Exploração: *High* (alta)
2. Nível de remedição: *Official Fix* (Consertada oficialmente)
3. Confidencia de reporte: *Confirmed* (confirmada)

1.4.3.2 Calculando a probabilidade

O cálculo da probabilidade normalmente é feito baseado em um banco de dados ou na experiência do usuário. De acordo com o PMBOK [36] é necessário estipular uma escala de probabilidade para que o risco possa ser calculado. Nossa escala vai ser baseada na escala do impacto que vai de 1 a 10, sendo 10 a maior probabilidade e 1 a menor probabilidade.

Nosso cálculo se baseia no banco de dados de testes, todas as vulnerabilidades serão verificadas através de testes, neste caso seria $P=F/E$. Onde P é a probabilidade, F é número de vezes que o teste falhou e E é numero de vezes que esse teste foi executado.

Para calcular a probabilidade é necessário o uso de um histórico dos casos de testes. Dois fatores podem ser utilizados: a experiência da equipe ou listas de vulnerabilidade que classificam as vulnerabilidades mais ocorridas na *web*, como por exemplo, o Top 10 da OWASP [37].

1.4.3.3 Análise de risco

Para analisar o risco do requisito pode ser utilizado a análise qualitativa proposta pelo PMBOK [36]. Para tal é verificado o risco baseado no impacto e probabilidade da vulnerabilidade acontecer. A tabela apresentada abaixo é mostrado a classificação do risco de acordo com a escala apresentada:

Tabela 3: Tabela de calculo de risco

Tabela de Riscos					
	Impacto				
	1 - 2	3 - 4	5 - 6	7 - 8	9 - 10

Probabilidade					
9 – 10	5	9	18	36	72
7 – 8	4	7	14	28	56
5 – 6	3	5	10	20	40
3 – 4	2	3	6	12	24
1 – 2	1	1	2	4	8

Legenda:

	Baixo
	Moderado
	Alto

Para os requisitos que tem mais de uma vulnerabilidade associada deve ser considerada a vulnerabilidade que tem o maior risco.

1.4.4 Testes de requisitos

Nesta seção é apresentado como validar os requisitos escritos que podem ser feitos através de duas técnicas:

1. Revisão dos requisitos;
2. Teste de requisitos.

A primeira técnica se baseia no processo formal de revisão de documentos, como mostrado na figura abaixo:

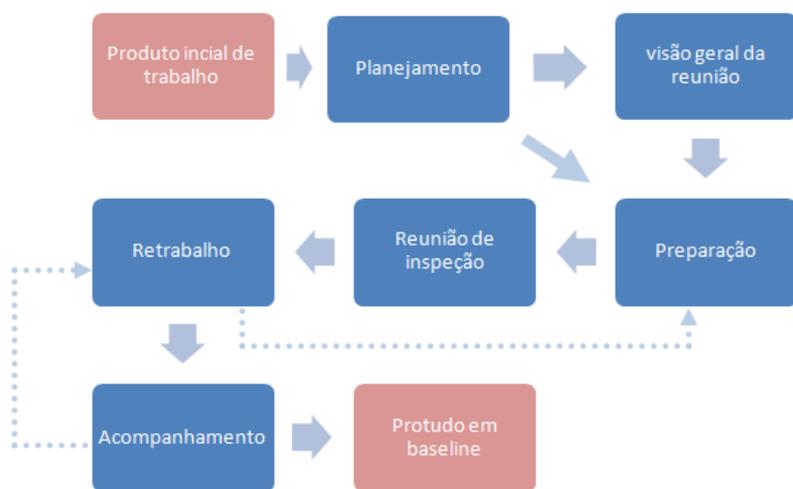


Figura 19 Processo de inspeção de requisitos [38]

Para ajudar na inspeção, um *checklist* foi proposto por [38] dividido em categorias.

Nos requisitos de segurança, é incluído mais uma categoria:

- Vulnerabilidades
 - I. As vulnerabilidades associadas aos requisitos foram listadas?
 - II. O impacto de cada vulnerabilidade foi calculado?
 - III. As ações de mitigação foram propriamente descritas?

A outra técnica de testes de requisitos é escrever pelo menos um caso de teste para todos os requisitos, verificando assim se o mesmo é testável. Pontos a serem considerados na escrita do caso de teste:

1. O identificador do requisitos – selecionar o requisito que vai ser testado por seu identificador, pelo menos um requisito tem que estar representado;
2. Requisitos associados - requisitos que podem ser relevantes para a escrita do caso de teste.
3. Descrição do teste – escrever qual o objetivo do teste e onde ele pode ser aplicado;
4. Problemas dos requisitos – problemas encontrados nos requisitos que fazem com que a escrita do teste seja difícil ou impossível.
5. Comentários e recomendações - para a solução dos problemas de requisitos encontrados.

É importante que os requisitos de segurança sejam escritos em forma de testes para cada vulnerabilidade associada ao requisito.

1.4.5 Testes de segurança

Nesta seção é mostrado como mapear, escrever, executar e automatizar testes de segurança. De acordo com [39], a técnica mais utilizada de testes de segurança são testes de penetração e a utilização de *patch's*. Os problemas com essas técnicas são que a primeira necessita de profissionais com alto nível de conhecimento em segurança e os *patch's* podem introduzir novas vulnerabilidades.

1.4.6 Planejando os testes

Os testes devem começar o mais cedo possível. Por isso, o planejamento dos testes não deve considerar sua execução só no final. Para tal a estratégia a seguir é utilizada:

1. Revisão do documento de requisitos – os requisitos devem ser mensuráveis, claros e testáveis.
2. Revisão do código – levando em consideração os pontos que precisam ser checados descritos no guia de segurança. As ferramentas utilizadas para esta técnica será escolhida de acordo com a tecnologia escolhida para escrever a aplicação. Para isso é utilizado um *checklist* de boas práticas da linguagem e codificação do sistema, ferramentas estáticas de análise de código e ferramenta de análise de vulnerabilidades.
3. Execução manual dos casos de testes;
4. Execução automática dos casos de testes.

Existem técnicas de testes estáticas e dinâmicas. As técnicas estáticas utilizadas nesse trabalho são as revisões de documento:

1. Revisão de documento de requisitos

2. Revisão de código utilizando ferramentas estáticas

Para as técnicas dinâmicas de testes vamos utilizar:

1. Automatização de testes funcionais;
2. Testes de penetração.

1.4.7 Projeto de testes: especificando casos de testes

Após o mapeamento das vulnerabilidades nos cenários, é feita uma seleção dos casos de testes. Todos os cenários e vulnerabilidades devem ser testados, mas não tendo tempo para tal, a seleção dos casos de testes é feita baseado no nível do impacto e na probabilidade de acontecimento da falha.

A escrita dos casos de testes segue um *template* definido de maneira similar considerando os seguintes pontos:

Tabela 4: *Template* de caso de testes formal

Campos	Descrição
ID	<<Identificador do caso de teste>>
Objetivo	<<Descrever qual o objetivo do caso de teste>>
Dados de entrada	<<Quais são os dados de entrada necessários para o teste>>
Passo	<<Quais são os passo para a execução do teste>>
Resultados esperados	<<Qual o resultado esperado para cada passo executado>>
Pré-Condições	<<Em que estado o sistema deve estar para execução do teste>>

Para escrita dos casos de testes de segurança outros pontos devem ser considerados:

1. No item de Objetivos deve ser definido qual o ataque será executado;
2. Inserir o item de vulnerabilidades que fazem parte do ataque
3. Determinar onde a segurança pode ser vulnerável
4. Qual o ambiente proposto do ataque, deve ser descrito nas pré-condições;
5. Quais ferramentas podem ou devem ser utilizadas para o ataque;
6. Quais *scripts* e massa de dados devem ser utilizados para o ataque.

O *template* do caso de testes de segurança, incluindo os campos acima apresentados, ficam da seguinte forma:

Tabela 5: *Template* de casos de testes de segurança

<<Identificador do caso de teste>> – <<Descrever qual o ataque que vai ser utilizado do caso de teste>>	
Severidade: <<Alta / Média / Baixa>>	Tipo de Execução: << Manual / Semi-Automática / Automática>>
Requisito:<< Identificador do requisito>>	Vulnerabilidade: << Nome da Vulnerabilidade a ser testada>>
Pré-Condições: <<Em que estado o sistema deve estar para execução do teste>>	
Dados de entrada: <<Quais são os dados de entrada necessários para o teste>>	
Passos:	Resultados esperados:
<<Quais são os passo para a execução do teste>>	<<Qual o resultado esperado para cada passo executado>>
Ferramenta: <<Qual a ferramenta utilizada>>	Script: <<Qual Script deve ser utilizado nesse teste>>

Para que os casos de testes sejam reusáveis é proposto que os mesmos sejam escritos em alto nível, e com objetivos amplos. Para uma rastreabilidade mais fácil do testes, eles são catalogados da seguinte forma:

Tabela 6: *Template* de rastreabilidade dos requisitos x vulnerabilidades

Requisito	Vulnerabilidade	Ferramenta	Caso de teste	Dados
RF<<xx>>	V1 - <<nome da vulnerabilidade>>	F1 –nome da ferramenta necessária	CT 001<<Id do caso de teste>>	Script o1<<Dados ou arquivos ou script necessários para execução >>
			CT 002	Dados.txt
		Teste manual	CT 004	NA
			CT 005	NA

	V2 - <<nome da vulnerabilidade>>			
RF XX	V3 <<nome da vulnerabilidade>>	Ferramenta	Casos de testes	Scripts, massa de dados ou arquivos

Assim fica fácil a localização dos casos de testes reusáveis para requisitos que possuem vulnerabilidades iguais ou reutilização dos casos de testes para outros sistemas a serem testados. Os *scripts* de testes serão escritos de forma parametrizada para que sua adaptação para outros sistemas seja mínima.

1.4.8 Reportagem dos resultados e análise dos testes

A reportagem de erros é uma fase extremamente importante para o processo de testes, pois neste ponto são identificados os problemas do sistema e baseado neles, pode-se fazer uma análise da qualidade do sistema, que é um dos fatores para a liberação ou não da *release* do mesmo.

1.4.8.1 Reportando Falhas

A reportagem das falhas precisa ser precisa e fornecer a maior quantidade de informações possíveis para o que o desenvolvedor possa reproduzir a falha reportada e ter dados suficientes para entender que pontos precisam ser corrigidos. Antes de reportar um erro é preciso seguir as instruções abaixo [33]:

- Verifique se este é realmente uma falha;
- Verifique se não é uma falha já conhecida;
- Determinar um conjunto de passos confiáveis para reproduzir a falha;
- Documente quaisquer informações adicionais que o ajudarão a reproduzir a falha

Após recolher todas as informações necessárias para reportar um erro de segurança, alguns pontos, além do normal, precisam ser reportados.

1.4.9 Template

Para reportar um erro de segurança encontrado é importante que os seguintes campos sejam preenchidos [105]:

Tabela 7 Template do relatório de falhas

Campos	Descrição
Repórter:	<<Quem reportou o erro>>
Severidade:	<< Qual a severidade do erro>>

Descrição	<<Uma breve descrição do erro>>
Ambiente:	<<Qual o ambiente foi utilizado no teste. Por exemplo: Browser IE 7.0.5730.13>>
Build:	<<Qual a baseline do sistema foi testado>>
Caso de Teste:	<< Qual o caso de teste achou o erro>>
Pré-condição:	<< Qual a pré-condição para executar o teste. Por exemplo: é necessário está logado como usuário xy no sistema>>
Passos:	<< Quais passos foram executados para atingir o erro>>
Resultado esperado:	<< Qual o resultado esperado se não houvesse o erro>>
Resultado atual:	<< Qual o resultado atual>>
Ferramenta:	<< Qual a ferramenta utilizada>>
Impacto:	<< Qual o impacto do erro>>
Ação de mitigação ou solução:	<<Qual a ação de mitigação prevista ou solução>>
Informações adicionais:	<< toda informação que possa ajudar o desenvolvedor entender e reproduzir o erro, como: print screen da tela, logs, scripts utilizados, dados de entrada entre outros >>

Para reportagem do erro não é necessária uma ferramenta especial para reportar os erros de segurança.

1.4.10 Análise dos erros reportados

É necessário que após os erros serem reportados, um análise seja feita para determinar o quanto urgente é o conserto. Para esta análise alguns pontos devem ser considerados:

Risco - O ativo do sistema está em risco devido a esta falha de segurança?

Custo do conserto - Qual é o custo de tempo para corrigir e fazer qualquer re-testes do código se essa falha for consertada?

Custo da exploração - Quais são os custos monetários e não monetários para a empresa, se a falha for utilizada em uma exploração de segurança? Quais são os custos para os clientes?

Efeito para Sistemas Dependentes - Quais são os efeitos sobre os sistemas dependentes caso a falha não seja consertada? Quais são os efeitos sobre os sistemas se a falha for consertada?

Após responder essas perguntas é mais fácil entender qual a urgência de consertar ou não uma falha reportada. Para responder as perguntas acima apresentadas, é utilizado como referência o documento de requisitos e o guia para saber qual o impacto, risco e perda que o sistema vai sofrer se a falha não for resolvida.

1.5 Conclusão

Computação nas nuvens (Cloud Computing), consiste de um conjunto de tecnologias que agrupadas trazem vantagens significativas para a gerenciamento dos ambientes de Tecnologia da Informação, conseqüentemente a diminuição dos custos.

No entanto, a idéia de que qualquer aplicação pode ser migrada para a nuvem sem a necessidade de alterações, consiste de uma visão simplista do problema. Esta afirmativa, é verdadeira desde que os gestores de tecnologia abdicuem das questões relacionadas a performance (no caso de computação nas nuvens o crescimento elástico) e segurança, caso estejam em execução em ambientes públicos.

Este capítulo de livro apresentou como se deve ser tratado tais questões em relação a aplicações existentes ou em desenvolvimento de forma que as mesmas possam atender os novos desafios colocados para as aplicações que devam estar em execução neste ambiente.

1.6 Bibliografia

- [1] Batten, C., Barr, K., Saraf, A., and Trepetin, S. pStore: A secure peer-to-peer backup system. Unpublished report, MIT Laboratory for Computer Science, (2001), 130-139.
- [2] Cox, L., Murray, C., and Noble, B. Pastiche: Making backup cheap and easy. ACM SIGOPS Operating Systems Review 36, (2002), 285-298.
- [3] Kubiatiowicz, J., Bindel, D., Chen, Y., et al. Oceanstore: An architecture for global-scale persistent storage. ACM SIGARCH Computer Architecture News 28, 5 (2000), 190-201.
- [4] Landers, M., Zhang, H., and Tan, K. PeerStore: better performance by relaxing in peer-to-peer backup. Proceedings of the Fourth International Conference on Peer-to-Peer Computing, (2004), 72-79.
- [5] Lillibridge, M., Elnikety, S., Birrell, A., Burrows, M., and Isard, M. A cooperative internet backup scheme. Proceedings of the USENIX Annual Technical Conference, (2003), 29-42.
- [6] Manber, U. and others. Finding similar files in a large file system. Proceedings of the USENIX Winter 1994 Technical Conference, October (1994), 1-10.
- [7] Muthitacharoen, A., Morris, R., Gil, T., and Chen, B. Ivy: A read/write peer-to-peer file system. usenix.org, .
- [8] Plank, J. A tutorial on Reed-Solomon coding for fault-tolerance in RAID-like systems. Software Practice and Experience 27, 9 (1997), 995-1012.
- [9] Rodrigues, R. and Liskov, B. High availability in DHTs: Erasure coding vs. replication. Lecture Notes in Computer Science 3640, 2 (2005), 226-239.
- [10] Rowstron, A. and Druschel, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Lecture Notes in Computer Science, November 2001 (2001), 329-350.
- [11] Stoica, I., Morris, R., Karger, D., Kaashoek, M., and Balakrishnan, H. Chord: A scalable peer-to-peer lookup service for internet applications. Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, (2001), 160.
- [12] Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., and Stoica, I. Looking up data in P2P systems. Communications of the ACM 46, 2 (2003), 43.
- [13] Rabin, M. Fingerprinting by random polynomials. Technical Report, Center for Research, TR-15-18, Havard Aiken Computer Laboratory, 1981.

- [14] Muthitacharoen, A., Chen, B., and Mazieres, D. A low-bandwidth network file system. Proceedings of the eighteenth ACM symposium on Operating systems principles, (2001), 174-187.
- [15] W. Lin, D. Chiu, and Y. Lee, "Erasure code replication revisited," Proceedings of the Fourth International Conference on Peer-to-Peer Computing, Citeseer, 2004, 90-97.
- [16] Oliveira, M., Cirne, W., Brasileiro, F., and Guerrero, D. On the impact of the data redundancy strategy on the recoverability of friend-to-friend backup systems. Proceedings of the 26th Brazilian Symposium on Computer Networks and Distributed Systems (SBRC'2008), Rio de Janeiro, Brazil, (2008).
- [17] Karger, D., Lehman, E., Leighton, F., Levine, M., Lewin, D., and Panigrahy, R. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. 29th Annual ACM Symposium on Theory of Computing, (1997), 654-663.
- [18] M. Oliveira. OurBackup: A P2P backup solution based on social networks, MSc Thesis, Universidade Federal de Campina Grande, Brazil, 2007.
- [19] FIPS 180-1. Secure Hash Standard. U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, VA, Apr. 1995.
- [20] L. Cox and B. Noble, "Samsara: Honor among thieves in peer-to-peer storage," Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, p. 120-132.
- [21] F. Aidouni, M. Latapy, and C. Magnien, "Ten weeks in the life of an eDonkey server," Proceedings of HotP2P'09, 2009, pp. 1-5.
- [22] R. Buyya and S. Venugopal, "A Gentle Introduction to Grid Computing and Technologies," CSI Communications, vol. 29, 2005, pp. 9-19.
- [23] I. Foster and C. Kesselman, The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 2003.
- [24] K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," Software: Practice and Experience, vol. 32, 2002, pp. 135-164.
- [25] F. BERMAN, A. HEY, and G. FOX, Grid Computing: Making the Global Infrastructure a Reality, Wiley, 2003.
- [26] T.a. DeFanti, I. Foster, M.E. Papka, R. Stevens, and T. Kuhfuss, "Overview of the I-Way: Wide-Area Visual Supercomputing," International Journal of High Performance Computing Applications, vol. 10, 1996, pp. 123-131.
- [27] A. ABBAS, Grid Computing: Practical Guide To Technology & Applications, Hingham, Massachusetts: Charles River Media, 2003.
- [28] "Open Grid Forum," 2009.
- [29] M. Chetty and R. Buyya, "Weaving Computational Grids: How Analogous Are They with Electrical Grids?," Computing in Science and Engineering, vol. 4, 2002, pp. 61-71.
- [30] I. Foster, "What is the Grid? A Three Point Checklist," Argonne National Laboratory & University of Chicago, 2002.
- [31] Assad, Rodrigo; Katter, Tarciana; Ferraz Felipe; Meira, Silvio; Security Quality Assurance on Web Application. ICSEA 2011
- [32] TALUKDER, Asoke K.; CHAITANYA, Manish. ARCHITECTING SECURE SOFTWARE SYSTEMS. Auerbach Publications, 2008. ISBN-13: 978-1-4200-8784-0
- [33] OWASP, 2008, OWASP TESTING GUIDE 2008 V3.0. Disponível em: http://www.owasp.org/index.php/Category:OWASP_Testing_Project , ultimo acesso em: 06/05/2011
- [34] SCHUMACHER, Markus et all. SECURITY PATTERNS - INTEGRATING SECURITY AND SYSTEMS ENGINEERING. New York: John Wiley & Sons, 2006. ISBN-13 978-0-470-85884-4

- [35] Peter Mell, Karen Scarfone, Sasha Romanosky. The Common Vulnerability Scoring System (CVSS). NIST, 2007. Disponível em: <http://www.first.org/cvss/> , ultimo acesso em: 06/12/2008
- [36] PHILLIPS, Joseph. PMP – PROJECT MANAGEMENT PROFESSIONAL – GUIA DE ESTUDO. Rio de Janeiro: Editora Campus, 2004. ISBN:85-352-1410-0
- [37] As 10 vulnerabilidades de segurança mais críticas em aplicações WEB. Disponível em: http://www.owasp.org/images/4/42/OWASP_TOP_10_2007_PT-BR.pdf . Ultimo acesso em: 17/08/2008
- [38] WIEGERS, Karl E. MORE ABOUT SOFTWARE REQUIREMENTS: THORNY ISSUES AND PRACTICAL ADVICE. Redmond: Microsoft Press, 2006. ISBN:0735622671
- [39] GHOSH, Anup K., MCGRAW, Gary, 1998, An Approach for Certifying Security in Software Components, published 21st National Information Systems Security Conference, National Institute of Standards and Technology (NIST). Disponível em: <http://csrc.nist.gov/nissc/1998/proceedings/paperA4.pdf>. Ultimo acesso em: 22/10/2008

Capítulo

2

Interfaces para Aplicações de Interação Natural Baseadas na API OpenNI e na Plataforma Kinect

Almerindo N. Rehem^{1,2}, Celso A. S. Santos¹ e Marcus V. R. Andrade²

¹ Universidade Federal da Bahia {almerindo.rehem@gmail.com , saibel@dcc.ufba.br}

² Universidade Tiradentes {vinny@touchingtheair.org}

Abstract

This course intent to introduce developers to concepts of the framework OpenNI (Open Natural Interaction) and the development of Natural Interaction applications using the Kinect hardware being compliant with the pattern proposed by the framework, so the same code can be used with different hardware that is certified by the OpenNI framework project. Besides the framework, the middleware library NITE will also be used, that is shipped and licensed separately, but can be used for non-commercial proposes.

Resumo

Este curso tem intenção de introduzir desenvolvedores aos conceitos do framework OpenNI (Open Natural Interaction) e o desenvolvimento de aplicações de interação natural usando o hardware Kinect de forma aderente aos padrões propostos pelo framewok. Sendo assim, o mesmo código pode ser usado com outro hardware que também seja certificado pelo projeto OpenNI. Além do framework também será usada a biblioteca de middleware NITE, que é fornecida e licenciada separadamente, mas pode ser utilizada para fins não comerciais.

2.1. Introdução

A Interação Usuário Computador (IUC) é uma área multidisciplinar que envolve áreas da ciência da computação, psicologia, linguística, artes, dentre outras. O conhecimento sobre as limitações da capacidade humana, restrições e evoluções (em termos de

dispositivos, interfaces e poder de processamento) das tecnologias existentes devem ser levados em conta para oferecer aos usuários uma forma adequada para interagir com plataformas computacionais. A evolução recente das tecnologias para captura de interações, muitas vezes de forma ubíqua, dos usuários criou novos paradigmas para a concepção de interfaces para aplicações interativas. O *Wiimote* [Wii Remote 2009] e o *Kinect* [Kinect (2010)], [Projeto OpenKinect 2011], [PrimeSense 2011] são os dois principais ícones comerciais que caracterizam essa nova forma de se pensar em interfaces, facilitando a comunicação entre usuário e computador por meio de interações naturais. Neste capítulo, considera-se como interação natural, uma comunicação entre o usuário e uma aplicação computacional realizada por gestos e/ou voz, em contrapartida à forma convencional baseada em janelas, ícones, *mouse* e apontadores (paradigma WIMP). O material apresentado neste texto é baseado na documentação oferecida aos desenvolvedores pela organização OpenNI disponível em [OpenNI User Guide 2011].

Lançado em 2006, o *Wiimote* (ou *Wii Remote*), baseado em acelerômetros, é o dispositivo principal de interação do console Nintendo Wii. O *Kinect*, fabricado pela *PrimeSense*, é o dispositivo principal de interação com o console *Xbox 360*, lançado em 2010 pela *Microsoft*, que traz como grande inovação uma interface baseada no reconhecimento de gestos. Apesar do foco inicial na área de jogos interativos, pesquisadores e desenvolvedores têm utilizado estas plataformas como base para a construção de interfaces naturais para aplicações interativas [Shiratori et al 2008], [Nakra et al 2009]. No caso do *Kinect*, em particular, menos de uma semana após seu lançamento oficial, a empresa *Adafruit* lançou o desafio “*Open Kinect Challenge*”, que oferecia US\$3.000 para o primeiro desenvolvedor de um *driver* de código aberto para o novo dispositivo [Adafruit Industries 2011]. Com o desafio e a recompensa lançados, não demorou para que a primeira implementação *opensource* de *drivers* para o *Kinect* (a biblioteca *libfreeneck* [Projeto OpenKinect 2011]) fosse apresentada, abrindo o caminho para o desenvolvimento de aplicações fora da área de jogos.

Reconhecendo a iniciativa e os esforços dos desenvolvedores espalhados pelo mundo, a *PrimeSense* fundou a *OpenNI Organization* com o objetivo de acelerar a introdução de aplicações de interação natural no mercado e, com isso, liberou o código fonte do seu *framework* para os desenvolvedores e pesquisadores. Esta sequência de fatos aliada à disseminação de vídeos com resultados experimentais na Web (principalmente através do *YouTube*) desencadeou uma corrida para a criação e desenvolvimento de aplicações para a plataforma *Kinect* e para o paradigma de Interação Natural baseado no reconhecimento de gestos.

Um dos grandes desafios para a área de IUC é a concepção de interfaces genéricas e de utilização simples dentro de um contexto que é muito influenciado por fatores tecnológicos e culturais. Um bom exemplo é a maneira como as janelas, ícones e menus direcionam um paradigma. Atualmente, as pessoas estão habituadas a acessar informações e serviços por meio de interfaces desenhadas baseadas em paradigmas WIMP ou em abstrações, como a Web. Um exemplo de metáfora desatualizada e ainda em uso é a utilização da figura de um disquete como ícone para salvar um arquivo. Outro exemplo é a interface típica de programas de reprodução de vídeos digitais com ícones baseados nas interfaces dos antigos videocassetes.

2.2. Interação Natural

A Interação Natural procura formas de tornar uma IUC o mais natural que possível, com objetivo de fazer com que as pessoas manipulem uma máquina sem perceber qual(is) artefato(s) é (são) utilizado(s) e sem necessariamente aprender um novo vocabulário correspondente ao conjunto de instruções desta interface.

Para [Heckel 1993], a interface deve procurar reproduzir a linguagem natural do usuário de computador ou da máquina, tentando prover o uso de palavras e expressões conhecidas, respeitando o vocabulário do usuário. Esta tentativa terá seus reflexos na redução do uso da memória e na diminuição do esforço cognitivo do usuário e, conseqüentemente, influirá numa interação mais “agradável” e natural.

[Valli 2007] define a interação natural em termos experimentais: as pessoas naturalmente se comunicam através de gestos, expressões, movimentos, além de explorarem o mundo real através da observação e manipulação de objetos físicos. E como uma tendência cada vez mais forte, as pessoas querem interagir com a tecnologia da mesma forma como lidam com o mundo real no cotidiano. A criação de novos paradigmas de interação e padrões alternativos de mídia que explorem as novas capacidades dos sensores presentes nas máquinas, e ainda, respeitem a espontaneidade inerente ao modo como o ser humano descobre e interage com o mundo é o principal desafio da construção de novas interfaces para interação. A tendência de quebra do paradigma tradicional das interfaces é evidenciada através de diversos experimentos usando novas propostas de interação surgindo cotidianamente na Web [Kinecthacks 2011] e do interesse de empresas como a *Microsoft*, que começa a demonstrar interesse na exploração dessas novas formas de interação [Toyama 1998]. Dessa forma, os usuários têm sido cada vez mais direcionados a um cenário de computação ubíqua, isto é, um cenário no qual os serviços digitais (ou computacionais) estarão disponíveis às pessoas em qualquer lugar e acessíveis para que interajam de forma natural com esses serviços [Weiser and Brown 1995], [Cordeiro Jr. 2009], [Abowd and Mynatt 2000].

[Saffer 2009] diz que um gesto na interação natural usuário computador é qualquer movimento físico que um sistema digital possa reconhecer e ao qual se possa responder. Um som, um inclinar de cabeça ou até mesmo uma piscada podem ser considerados gestos. Sendo o reconhecimento dos gestos um dos problemas atuais no desenvolvimento de aplicações dependentes de contexto, o poder de se contextualizar para poder reconhecer os gestos de forma precisa e responder da mesma forma é o que possibilita a interação natural usuário computador.

[Gallud *et all* 2010] define a comunicação não verbal como aquela que utiliza pistas e sinais sem estrutura sintática e separa essa comunicação em três tipos:

- **Paralinguagem:** Refere-se aos elementos que acompanham expressão linguística, consistindo de pistas e sinais que sugerem uma interpretação diferente do que está sendo dito, percebe-se pela intensidade e/ou altura da voz do emissor bem como chorar, rir, controle respiratório, etc.
- **Linguagem corporal:** Caracteriza-se pela utilização de gestos, porém seguida de características distinguíveis utilizando o tronco, extremidade e cabeça também pela proximidade entre emissor e receptor, expressões de estado como felicidade, honestidade, cumplicidade, etc.

- Linguagem sonora: Acompanha os gestos e caracteriza-se pela interpretação qualquer som que expresse emoção, contextualiza uma cena e, até mesmo o silêncio pode ter significado em uma mensagem.

A detecção e interpretação de tantos sinais para uma resposta adequada dependem de análise e contextualização extremamente rápidas dos sinais recebidos. Assim, inúmeros fatores podem influenciar o entendimento correta de uma mensagem na comunicação usuário computador.

Se for possível utilizar linguagem corporal e falada como intervenções de usuário, abre-se a possibilidade de criar interfaces onde o usuário e a máquina poderão se comunicar de forma mais humana e natural. Por outro lado, a criação de interfaces baseadas na comunicação usuário computador por meio de gestos e voz traz consigo um novo tipo de problema: quais gestos de interação e comandos associados devem ser utilizados para realizar uma determinada tarefa?

A proposta deste capítulo é apresentar uma forma padronizada de desenvolver de forma modular os componentes, que somados viabilizam a concepção de aplicações de interação natural, podendo ser compartilhados e reutilizados para aplicações de diferentes finalidades e possibilitando a criação de um ecossistema de bibliotecas e aplicações para auxiliar os desenvolvedores da área em questão.

2.3. Evolução dos Dispositivos de Interação Usuário Computador

Esta seção do texto irá abordar alguns dos principais dispositivos que mudaram a forma de se pensar a interação usuário computador, finalizando com a apresentação da arquitetura do *Kinect* e os seus recursos associados.

2.3.1. Os dispositivos Wiimote e Playstation Move

O *Wiimote* e o *Playstation Move* se parecem muito em termos de funcionamento. Eles são baseados essencialmente em emissão de luz infravermelha e uma câmera de RGB tradicional. A luz infravermelha é detectada pela câmera RGB e assim, é possível detectar a posição do ponto de referência: no caso do *Wiimote* esse ponto é o próprio controle e no caso do *Playstation Move*, é a peça com uma extremidade luminosa que melhora a precisão que é dada pela câmera RGB. Este processo pode ser feito de forma 2D resultando numa posição na tela, exatamente como um ponteiro de mouse, ou de forma 3D, onde para detectar a distância do controle até a tela é utilizada uma triangulação. Além disso, quando há um acelerômetro disponível ao conjunto de sensores, além de posição e distância, as inclinações em qualquer eixo também podem ser detectadas. Essa inclinação tem o nome de *DOF* (*Degrees Of Freedom*). Estas técnicas, já eram conhecidas desde a década de 80 quando a Nintendo lançou o controle *Power Glove* (Figura 1.1) que funcionava de maneira muito similar aos dispositivos *Wiimote* e *Playstation Move*, com a diferença que, em conjunto com a câmera RGB, ao invés de infravermelho, utilizavam-se sensores de ultrassom para medir o tempo entre a reflexão das ondas emitidas por um aparelho posicionado na TV e o material da luva.

Além das coordenadas (x,y,z) do controle em relação a cena, para uma boa experiência também é necessário que os seus ângulos (*roll*, *pitch* e *yaw*) possam ser detectados possibilitando calcular a inclinação (ver Figura 1.2). Tanto o *Playstation Move* quanto o *Wiimote* junto com o *Motion Plus* já trabalham com esses 6 graus de liberdade. A *Power Glove*, ainda na década de 80, já retornava além das coordenadas

(x,y,z) , também fornecia o *yaw* e o *roll*. Apesar da quantidade enorme de informações sobre posicionamento e inclinação obtidos com esses dispositivos, ainda assim são poucas as informações para imitar o mundo real, simplesmente porque as informações estão relacionadas apenas ao controle e por isso, não levam em conta a cena ou o jogador/ator em si, nem permite a detecção de gestos.

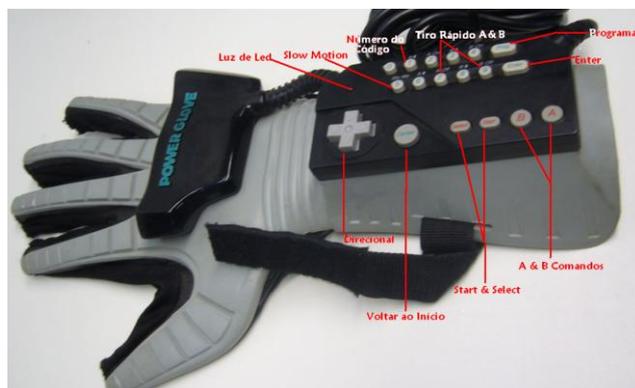


Figura 1.1. Nintendo *Power Glove*, um dos antecessores do *Wiimote*.
Fonte: [Power Glove 2008]

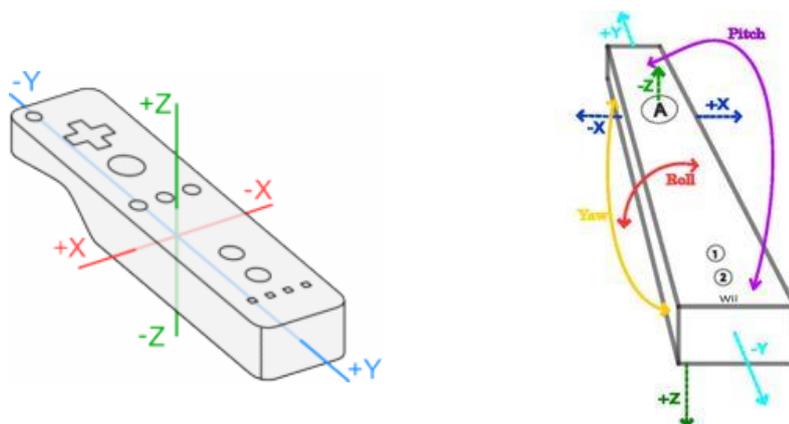


Figura 1.2. (a) Eixos de referência do *Wiimote* e (b) os seis graus de liberdade.
Fontes: [Kyma 2010] e [Wiimote 2010]

No caso do *Kinect*, a geração de um mapa de profundidade (cada pixel da imagem está relacionado a uma distância do pixel até a câmera) a qual é a principal inovação, possibilita ao dispositivo entregar informações 3D da cena completa, incluindo o jogador/ator e não somente do controle do usuário. Para se conseguir este resultado, a tecnologia por trás do *Kinect* utiliza o conceito de luz estruturada: uma luz com um padrão de projeção bem definido que permite o cálculo das informações da cena partindo da combinação entre os padrões do que é projetado e da distorção da projeção em função dos obstáculos encontrados pela luz (ver Figura 1.3).

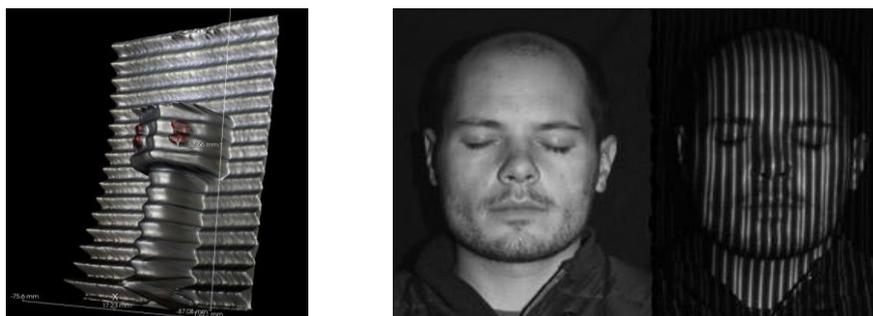


Figura 1.3. Luz estruturada (a) projetada em superfície irregular e (b) usada no mapeamento da face.

Fonte: [Shape Quest 2010]

2.3.2. O Kinect

O *Kinect* possui um sofisticado algoritmo de processamento paralelo (embarcado no chip SoC) necessário para extrair o mapa de profundidade a partir da luz estruturada recebida (Figura 1.4 e Figura 1.5). Para possuir mais precisão nas informações dos sensores, as imagens são alinhadas pixel a pixel, ou seja, cada pixel de imagem colorida é alinhado a um pixel da imagem de profundidade. Além disso, o *Kinect* sincroniza (no tempo) todas as informações dos sensores (profundidade, cores e áudio) e as entrega através do protocolo USB 2.0 [Crawford 2010].

A Figura 1.4 e a Figura 1.5 apresentam, respectivamente, o posicionamento dos sensores e o diagrama esquemático do *Kinect*. Nas figuras, é possível identificar como os sensores estão conectados e quais são os fluxos de dados gerados pelo dispositivo.

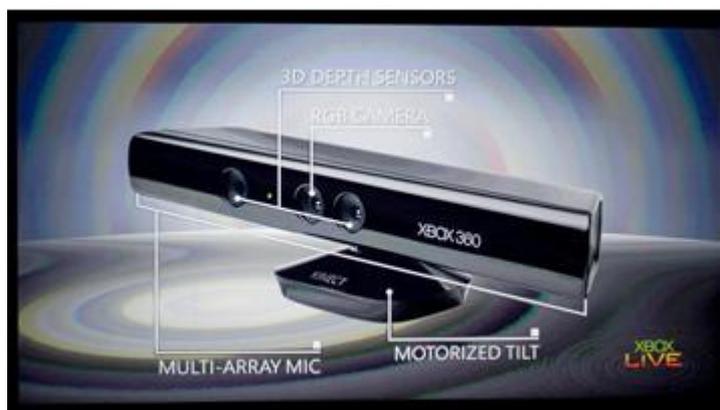


Figura 1.4. (a) Sensores do *Kinect* no XBOX 360.

Fonte: [Crawford 2010]

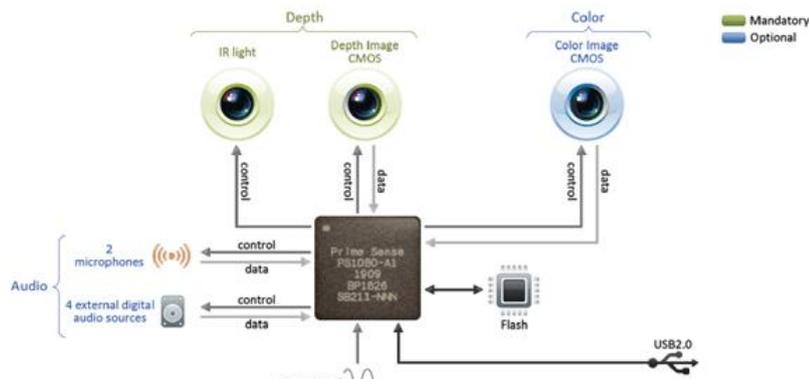


Figura 1.5. Diagrama de blocos do Kinect na especificação da PrimeSense.
Fonte: [PrimeSense 2011]

O Kinect dispõe de vários recursos (som, imagem, profundidade, infravermelho, motor de movimentação) com um alto índice de precisão e sincronismo em um único dispositivo. Estes recursos oferecem uma série de possibilidades de interação inovadoras entre usuários e serviços e aplicações computacionais. O *framework OpenNI* e o módulo *NITE*, tratados nas próximas seções do texto, permitem construir interfaces de aplicações sofisticadas, em um nível mais alto de abstração, tendo como base a combinação de recursos da plataforma Kinect. As seções seguintes do texto são fortemente baseadas na recente documentação [OpenNI User Guide 2011], um guia para usuários desenvolvedores interessados em utilizar o *framework OpenNI*.

2.4. Introdução ao OpenNI

OpenNI (ou *Open Natural Interaction*) designa uma instituição sem fins lucrativos e também uma marca registrada da PrimeSense [PrimeSense 2011]. A *OpenNI* busca promover a interoperabilidade e compatibilidade entre aplicações, *middleware* e dispositivos de interação natural, atuando na verificação e certificação da conformidade das soluções de diferentes fabricantes e desenvolvedores aos padrões definidos pelo *framework OpenNI*.

O principal propósito da instituição *OpenNI* é especificar uma API padrão para a comunicação tanto para dispositivos (sensores) quanto para aplicações e *middlewares*. Com isso, busca-se quebrar a dependência entre os sensores e os *middlewares*, além de permitir o desenvolvimento e portabilidade de aplicações para diferentes módulos de *middlewares* com menor esforço adicional (conceito “*write once, deploy everywhere*”). Mais ainda, a API fornece: (i) o acesso direto aos dados brutos dos sensores aos desenvolvedores e (ii) a possibilidade de se construir dispositivos que funcionem em qualquer aplicação compatível com o padrão proposto aos fabricantes.

O *framework OpenNI* habilita aos desenvolvedores acompanharem as cenas do mundo real (cenas em 3D), utilizando tipos de dados processados capturados por um sensor de entrada. Por exemplo, a representação do contorno de um corpo humano, a localização de uma mão na tela, etc. Essas aplicações, graças ao padrão *OpenNI*, são construídas com um alto nível de desacoplamento, ou seja, podem ser construídas de forma independente dos fabricantes de sensores ou de um determinado *middleware*.

A Figura 1.6 oferece uma visão abrangente, em camadas, da arquitetura *OpenNI* e dos conceitos associados a esse padrão. Essas camadas são divididas em *Top*, *Middle*, *Bottom*, e descritas da seguinte forma:

- **Top:** Representa os softwares que implementam aplicações de Interação Natural seguindo o padrão *OpenNI*;
- **Middle:** Representa o próprio *framework OpenNI* e oferece interfaces de comunicação para interagir tanto com os middlewares quanto com os dispositivos (sensores);
- **Bottom:** Ilustra a camada de mais baixo nível, onde se situam os dispositivos de hardware responsáveis por capturar elementos visuais e auditivos da cena.

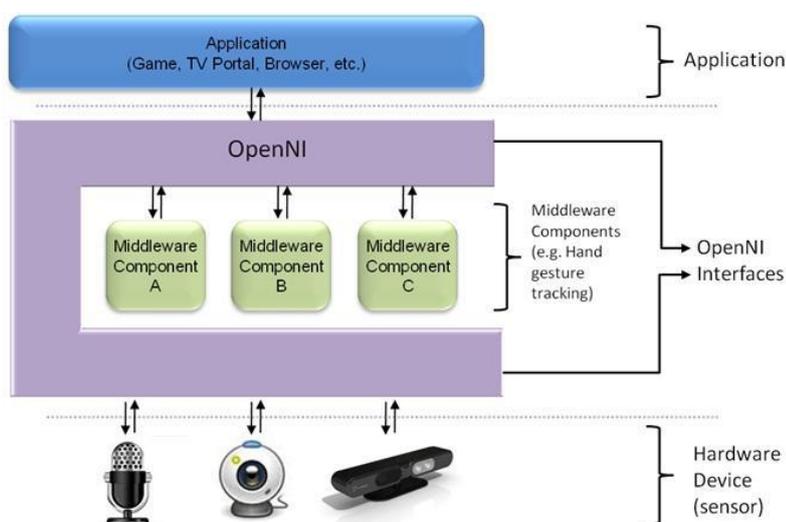


Figura 1.6. Abstração e componentização do *OpenNI*.
Fonte: [PrimeSense 2011]

2.4.1. O Framework OpenNI

Pelo exposto até aqui, pode-se afirmar que o *framework* é uma camada de abstração que fornece interfaces para os componentes de softwares (*middlewares*) e para os dispositivos físicos através de uma API. Esta API habilita registros de múltiplos componentes (módulos) no *framework* que são responsáveis por produzir ou processar dados dos sensores físicos de forma flexível. Atualmente, os componentes suportados pela arquitetura são agrupados em dois tipos, os módulos de sensores e *middlewares*, descritos como se segue:

- **Módulos de sensores:**
 - **Câmera RGB:** dispositivo responsável por gerar imagens coloridas;
 - **Sensor 3D:** dispositivo capaz de gerar o mapa de profundidade;
 - **IR câmera:** dispositivo de infravermelho;
 - **Dispositivo de áudio:** um ou mais dispositivos de microfones
- **Middlewares:**
 - **Middleware de análise de corpo:** componente de software capaz de processar os dados de entrada dos sensores e retornar informações relacionadas ao corpo humano. Como por exemplo, uma estrutura de

dados que descreva as articulações, orientação e o centro de massa de uma pessoa na cena.

- **Middleware de análise de mão:** componente capaz de processar dados dos sensores, reconhecer e retornar uma coordenada de localização do centro de uma mão em cena.
- **Middleware de detecção de gestos:** componente capaz de identificar gestos pré-definidos e alertar a aplicação todas as vezes que houver ocorrência desses gestos.
- **Middleware de análise de cena:** componente capaz de analisar os dados da cena e produzir informações como: (i) a separação dos planos *foreground* e do *background*; (ii) as coordenadas do chão e (iii) a identificação de atores em cena.

A Figura 1.7 ilustra um exemplo de aplicação dos componentes da arquitetura. No exemplo, é exibido um cenário em que cinco módulos são registrados para trabalhar em uma instância do *framework OpenNI*. Dois dos módulos registrados são sensores 3D conectados fisicamente ao host. Os outros três, são componentes de *middleware*, sendo que os dois primeiros módulos retornam dados relacionados ao corpo humano e o último, dados relacionado à mão (*hand point*).

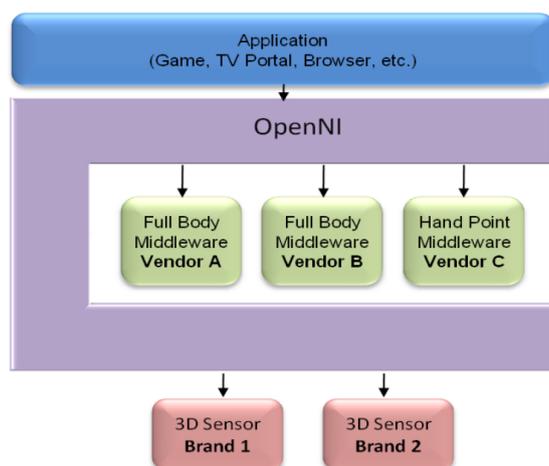


Figura 1.7. Exemplo de uso da arquitetura onde são registrado cinco módulos simultâneos em uma única instância do *OpenNI*.

Fonte: [PrimeSense 2011]

2.4.2. Nós de produção

Para um melhor entendimento da arquitetura *OpenNI*, é necessário compreender o conceito de base desta arquitetura, denominado nó de produção (*production node*). Todo componente que produza dados necessários para as aplicações baseadas em interação natural é chamado de nó de produção. Um nó de produção pode utilizar outros nós, ser usado por nós de mais alto nível ou diretamente pela própria aplicação.

Os nós de produção são divididos em duas categorias: (i) nós relacionados ao *middleware* e (ii) nós relacionados aos sensores.

- **Nós relacionados aos sensores:**
 - **Device:** nó de produção que representa um dispositivo físico (sensor de profundidade, ou uma câmera RGB) e tem como principal objetivo permitir configurações dos dispositivos;

- **Depth Generator**: nó responsável por gerar um mapa de profundidade;
 - **Image Generator**: nó responsável por gerar um mapa de pixels RGB (imagem colorida);
 - **IR Generator**: nó que gera o mapa de imagem infravermelha;
 - **Audio Generator**: nó responsável pelo tratamento da *stream* de áudio.
- **Nós relacionados ao *middleware***:
 - **Gestures Alert Generator**: nó de produção responsável por alertar a aplicação quando houver um reconhecimento de um gesto;
 - **Scene Analyzer**: nó responsável por gerar os dados de análise de cena, possibilitando a separação das camadas *foreground*, *background*, *floor plane* e identificação do ator;
 - **Hand Point Generator**: nó responsável pelo suporte à detecção e rastreamento de uma mão (esse nó produz um sinal quando uma mão é detectada ou quando sua localização é alterada);
 - **User Generator**: nó responsável por produzir uma representação parcial ou total do corpo humano em uma cena.
 - **Recorder**: nó que implementa a funcionalidade de gravação de dados;
 - **Player**: nó responsável por ler os dados gravados e exibi-los;
 - **Codec**: nó responsável pela compressão e descompressão dos dados.

Para facilitar o entendimento da utilização desses nós de produção e seus relacionamentos, pode-se usar um exemplo inicial de aplicação na qual é necessário acompanhar o movimento de uma figura humana na cena. Esta aplicação requer nós de produção que forneçam dados referentes ao corpo humano ou, em outras palavras, um nó *UserGenerator*. Por sua vez, o nó de produção *UserGenerator* necessita obter dados do nó *DepthGenerator* (outro nó de produção de um nível mais baixo na hierarquia), que é implementado para capturar os dados brutos de um sensor de profundidade (capturar N frames por segundo e retornar o mapa de profundidade). Essa seqüência de relacionamentos e dependências entre os nós de produção é conhecida pelo *OpenNI* como cadeia de produção (*Production Chain*) e será mais detalhada na próxima seção.

Outros exemplos comuns de aplicações com saída de dados com um alto nível de abstração são os casos de localização de uma mão, identificação de pessoa em cena e a identificação de gestos, descritos a seguir:

- Localização da mão de um usuário (Figura 1.8.a): o nó de produção pode retornar o centro da palma da mão através de um ponto (comumente referenciada pelo *hand point*), ou a localização das pontas dos dedos;
- Identificação de um ator na cena (Figura 1.8.b). o nó de produção retorna a localização momentânea e orientação das articulações do ator (comumente referenciada como *body data*);
- Identificação de gestos (como um aceno, na Figura 1.8.c): o nó de produção se responsabiliza por identificar e alertar a aplicação, através do *framework*, quando um determinado gesto ocorrer.

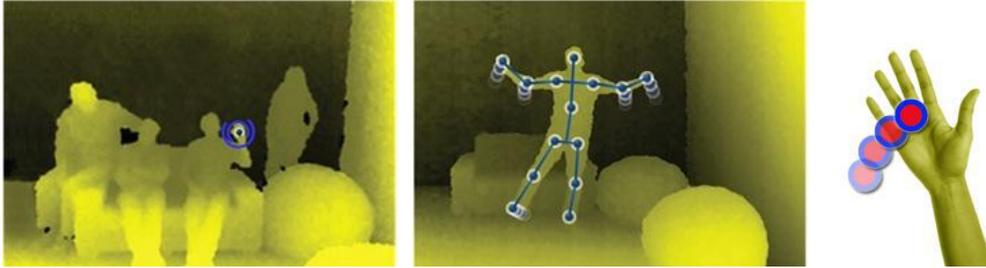


Figura 1.8. (a) Detecção da Mão; (b) Detecção do Corpo; (c) Detecção de Gestos.
Fonte: [OpenNI User Guide 2011]

2.4.3. As cadeias de produção

Na seção anterior foi apresentado um exemplo de aplicação que usa o nó *User Generator* para produzir a representação do corpo humano (*body data*). Este nó de produção *UserGenerator*, por sua vez, necessita de informações geradas por outros nós dos níveis inferiores da arquitetura. Os dados, relativos ao corpo humano, gerados pelo nó de produção *UserGenerator* são utilizados pela aplicação em si ou por outros nós de produção, definindo assim, a cadeia de produção da informação.

Uma aplicação comumente irá necessitar dos nós de produção de mais alto nível, tais como o *UserGenerator* e o *HandPointGenerator*. Essa necessidade é contemplada pela arquitetura, pois o *framework* possibilita a utilização dos nós de produção de mais alto nível sem que o desenvolvedor se preocupe com a cadeia de produção. Entretanto, o *OpenNI* é flexível o suficiente para permitir o acesso direto à configuração dos nós pertencentes às cadeias de produção.

2.4.4. As funcionalidades

As funcionalidades (tradução livre para *capabilities*) são mecanismos para permitir o registro de nós de produção de um mesmo tipo, mas produzidos por fabricantes diferentes. Com isso, uma aplicação pode “perguntar” ao *framework* quais dos nós do conjunto possuem a funcionalidade que ela necessita e, assim, incluir os nós necessários para a geração da cadeia de produção. Atualmente, as funcionalidades suportadas pelo *framework* são:

- **Alternative view:** habilita um sensor do tipo *Map Generator* a transformar seus dados para que pareça que o sensor tenha sido colocado em um local diferente do qual ele realmente se encontra;
- **Cropping:** possibilita que um sensor do tipo *Map Generator* gere dados apenas de uma área do frame ao invés de todo o *frame*;
- **Frame Sync:** habilita dois sensores de qualquer tipo a sincronizarem suas informações de modo que os dados cheguem aos seus nós consumidores ao mesmo tempo;
- **Mirror:** possibilita que sensores “visuais” informem seus dados de forma espelhada, de forma semelhante à formação da imagem num espelho;
- **Pose Detection:** permite que um nó do tipo *User Generator* reconheça quando um usuário está em uma posição específica;

- **Skeleton**: habilita que o *User Generator* forneça como dados de saída o esqueleto do usuário; permite também a localização das articulações, a identificação da posição do esqueleto e a calibração do usuário;
- **User Position**: habilita um sensor do tipo *Depth Generator* a otimizar a saída do mapa de profundidade gerado para uma área específica da cena;
- **Error State**: permite que um nó de qualquer tipo informe que ele está em um estado de erro ou que não está em perfeito funcionamento;
- **Lock Aware**: permite que um nó qualquer seja bloqueado para uso fora de um determinado contexto; em outras palavras, não permite que ele seja acessado fora de uma determinada cadeia de produção ou por um nó diferente daquele que efetuou o bloqueio.

2.4.5. Porque utilizar o *framework OpenNI*?

A organização e implementação de uma camada de abstração do *hardware* e a flexibilidade obtida com registros de *middlewares* tornam as soluções desenvolvidas dentro do padrão *OpenNI* mais abrangentes e compatíveis com um número maior de *hardwares* e de bibliotecas que possam vir a ser distribuídas.

No caso da disponibilização de códigos fonte ou de bibliotecas compiladas não aderentes ao *framework* existe a possibilidade de concepção de *Wrapper* para garantir a aderência e a utilização dentro do *framework*. Em último caso, é possível ainda acessar diretamente a biblioteca e fazer com que o nó construído seja o ponto gerador de informação proveniente da biblioteca.

Com o objetivo de posicionar o projeto *OpenNI* e apresentar as vantagens desse projeto com relação ao *Kinect SDK*¹ recentemente liberado pela *Microsoft* [Kinect SDK (2010)], três tabelas comparativas são apresentadas na sequência do texto. Os critérios levados em conta na comparação das duas opções de programação foram: (i) o licenciamento das bibliotecas e ferramentas; (ii) o suporte a *hardware* e (iii) as funcionalidades inclusas. Note que a vantagem incontestável do *OpenNI* com relação à portabilidade (suporte ao *Windows 7/Vista/XP*, *Linux* e *MacOS X*) sobre o *Microsoft Kinect SDK* (que funciona apenas para *Windows 7*) não foi considerada na comparação.

Tabela 1.1. Comparação quanto ao licenciamento.

	<i>OpenNI</i>	<i>Microsoft Kinect SDK</i>
Ferramentas para desenvolvimento	Uso comercial	Uso comercial
Bibliotecas e <i>framework</i>	Uso comercial	Uso não comercial

Tabela 1.2. Comparação quanto ao suporte a dispositivos de hardware.

	<i>OpenNI</i>	<i>Microsoft Kinect SDK</i>
Suporte de Áudio	Não Suporta	Suporta
Suporte ao motor de passos	Suporte Parcial ²	Suporta
Suporte ao sensor de profundidade	Suporta	Suporta
Suporte ao sensor RGB	Suporta	Suporta
Suporte a múltiplos sensores	Suporta	Suporta
Suporte a sensores de outros fabricantes	Suporta ³	Não suporta

¹ *Kinect SDK* (ou *Kinect Software Development Kit*) é o *kit* de desenvolvimento de software para o *Kinect* oferecido pela *Microsoft*.

² O suporte pode ser obtido usando um *driver* de outro equipamento, que é parcialmente compatível.

³ Qualquer *hardware* certificado *OpenNI*, aderente ao *framework*, é suportado.

Tabela 1.3. Comparação quanto às funcionalidades inclusas.

	<i>OpenNI</i>	<i>Microsoft Kinect SDK</i>
Acompanhamento de corpo inteiro	Incluso	Incluso
Acompanhamento apenas de mão	Incluso	Não incluso
Alinhamento automático das informações do sensor de profundidade e RGB	Incluso	Não incluso
Cálculo da posição das juntas	Incluso	Incluso
Cálculo da rotação das juntas	Incluso	Não Incluso
Sistema de reconhecimento de gestos	Incluso	Não Incluso
Eventos para reconhecimento de entrada e saída de novos usuários na cena	Incluso	Não Incluso
Eventos para cada frame disponível na <i>stream</i> dos sensores RGB e de profundidade	Não Incluso	Incluso
Suporte para gravação para o disco e reprodução a partir dele, das informações dos sensores	Incluso	Não Incluso

Concluindo, pode-se prever a continuidade de desenvolvimento e a incorporação de melhorias no curto prazo no *SDK* da *Microsoft*, uma vez que o *Kinect* é um projeto lançado pela própria empresa. Por outro lado, no momento da escrita deste documento, o *OpenNI* é um projeto que se encontra com um nível alto de maturidade e com um número importante de aplicações e desenvolvedores já consolidados. Além disso, o *OpenNI*, diferentemente do *SDK* da *Microsoft*, não se limita a fornecer aos desenvolvedores um conjunto de métodos, mas uma arquitetura padrão de desenvolvimento para aplicações na área de interação natural, abrangendo fabricantes de sensores, desenvolvedores de *middlewares* e de aplicações dessa área.

2.5. Programando com o *OpenNI*

As seções seguintes deste capítulo terão como foco principal os aspectos práticos para programação utilizando o *framework OpenNI*, fornecendo o conhecimento básico e essencial para qualquer desenvolvedor iniciar suas aplicações relacionadas à interação natural dentro desta plataforma. O texto está estruturado de forma a apresentar: (i) os principais objetos usados na programação; (ii) a integração entre o *OpenNI* e o *Eclipse CDT*⁴ [Eclipse CDT Project 2011] e (iii) alguns códigos comentados de exemplos de uso do *framework*. Por outro lado, não serão abordadas informações relacionadas à instalação do sistema operacional *Linux*, do *OpenNI* e do *middleware* NITE e suas bibliotecas requeridas. Estas informações podem ser encontradas em detalhes nas referências [PrimeSense 2011], [Projeto OpenKinect 2011], [OpenNI User Guide 2011].

2.5.1. Principais Objetos do Framework OpenNI

Os objetos mais comuns para o desenvolvimento de aplicações que utilizem o *framework OpenNI* são o *Context Object*, o *MetaData Object* e os objetos geradores de dados ou *Data Generators* (em especial, *Map Generator*, *Depth Generator*, *Image Generator*, *IR Generator*, *Scene Generator*, *Audio Generator*, *Gesture Generator*, *Hand Point Generator* e *User Generator*).

2.5.1.1. Objetos Context

A classe mais importante no *OpenNI* é a *Context* (Figura 1.9). O objeto pertencente a esta classe é a própria instância do *OpenNI*, sendo responsável por manter o estado

⁴ *CDT (C/C++ Development Tooling)* é um ambiente de desenvolvimento integrado para C e C++ baseado na plataforma Eclipse resultado do projeto CDT [Eclipse CDT Project 2011].

completo da aplicação, inclusive de todas as cadeias de produção utilizadas. Mais de um *Context* pode ser instanciado, entretanto as informações entre eles não podem ser compartilhadas. Por exemplo, um nó de produção do tipo *middleware* não consegue acessar informações de um nó do tipo sensor que esteja em um contexto diferente.



Figura 1.9. A Classe *Context* está no nível mais alto da hierarquia.

Para se iniciar o *framework*, o objeto *Context* deve ser instanciado em memória e todos os módulos necessários para aplicação que forem conectados ao *Context* serão carregados e analisados. O *Context* será então o ponto central de acesso a todos os módulos da aplicação e responsável também por liberar toda a memória utilizada pelo *OpenNI* através da função *shutdown()*, chamada pela aplicação.

2.5.1.2. Objetos Metadata

Os objetos *Metadata* (ver Figura 1.10) estão presentes em todos os nós de produção que gerem dados brutos (*Depth Generator*, *Image Generator*, *IR Generator*, *Scene Generator*, *Audio Generator*). Este objeto oferece um encapsulamento das propriedades relacionadas com os dados de quem ele pertence. Como por exemplo, a resolução (número de pixels nos eixos *x* e *y*) de um mapa de profundidade (*Depth Map*) obtido através do *DepthGenerator*.

Esse tipo de objeto seria necessário no caso de uma aplicação que tivesse que ler, a todo instante, os dados de um mapa de profundidade (*Depth Map*) que foi inicialmente configurado com resolução QVGA (320 x 240 *pixels*). Se, em algum momento, a resolução de saída fosse alterada para VGA (640 x 480 *pixels*), a aplicação teria uma inconsistência de informação até que novos *frames* fossem gerados. Na verdade, seria um erro assumir que a resolução encontra-se em VGA e tentar acessar *pixels* que não estivessem presentes na matriz (mapa de profundidade).

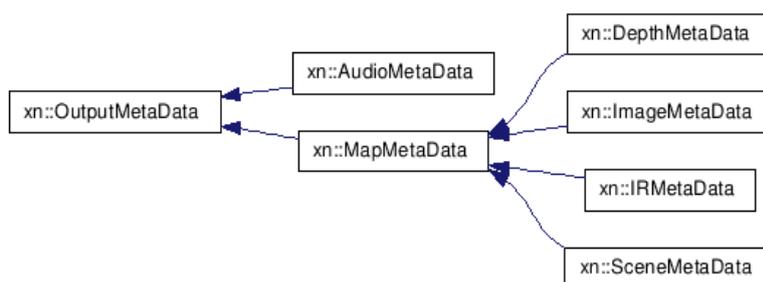


Figura 1.10. Hierarquia de Classes Metadata.

Como todos os *nós de produção geradores* possuem seu próprio *Metadata Object* responsável por registrar a propriedade dos dados. A solução do exemplo anterior seria obter esse *Metadata Object*, e a partir dele, ler os dados reais e sua resolução em tempo de execução (nesse caso, um mapa de profundidade QVGA).

2.5.1.3. Objetos *Data Generators*

Generator é um nó de produção que fornece dados de retorno para a aplicação. Se o nó retorna algum *map* (matriz de pixels) como resultado, então esse nó de produção implementa a interface básica *MapGenerator*. O diagrama da hierarquia dos objetos do tipo *Generator* é ilustrado pela Figura 1.11.

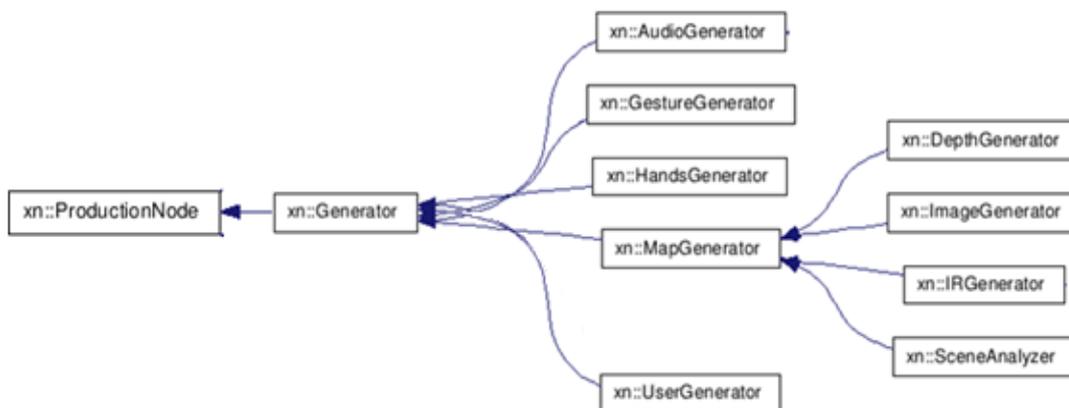


Figura 1.11. Hierarquia de classes dos *Generators*.

Os principais objetos e funções dos objetos do tipo *Generator* são os seguintes:

- ***DepthGenerator***: Nó *Generator* que tem como foco retornar um mapa de profundidade (matriz de pixels de profundidade). Suas principais funcionalidades são:
 - ***Get Depth Map***: retorna o *map* de profundidade;
 - ***Get Device Max Depth***: retorna a distância máxima disponível pelo dispositivo;
 - ***Field of View property***: configura o campo de visão (valores dos ângulos horizontais e verticais) do sensor;
 - ***User Position capability***: retorna a posição do usuário.
- ***ImageGenerator***: Nó *Generator* que retorna um mapa de imagem colorida (matriz de pixels coloridos) e tem como principais funcionalidades:
 - ***Get Image Map***: retorna o *map* da imagem;
 - ***Pixel Format Property***
- ***IRGenerator***: Nó *Generator* que fornece um mapa infravermelho e que tem como principal funcionalidade:
 - ***Get IR Map***: retorna o mapa de infravermelho atual
- ***SceneAnalyzer***: Um *Generator* que obtém dados brutos do sensor e retorna um *map* rotulando as camadas da cena. Suas principais funcionalidades são:
 - ***Get Label Map***: Fornece um *map* em que cada pixel é rotulado de acordo com a cena. Por exemplo, Ator1, Ator2, chão, e assim por diante.
 - ***Get Floor***: retorna as coordenadas do *floor plane* (chão da cena).
- ***AudioGenerator***: Objeto que retorna dados de áudio e possui como principal funcionalidade:

- **Get Audio Buffer;**
- **Wave Output Modes Properties:** Configura a saída do áudio, incluindo sua taxa de amostragem e bits por amostra.
- **GestureGenerator:** Objeto que possibilita o rastreamento de um corpo humano ou de uma mão em específico com o intuito de identificar um gesto. Suas principais funcionalidades são:
 - **Add/Remove Gesture:** Habilita ou desabilita um determinado gesto. Uma vez habilitado o nó irá observar toda vez que o gesto acontece.
 - **Get Active Gesture:** Retorna todos os gestos disponíveis
 - **Register/Unregister Gesture Callbacks:** Habilita ou desabilita funções de retorno. Estas funções são chamadas quando o gesto ocorrer.
- **HandsGenerator:** Objeto que permite o rastreamento de uma mão
 - **Start/Stop Tracking:** Inicia ou pára o rastreamento de uma mão;
 - **Register/Unregister Hand Callbacks:** Permite que funções de retorno sejam registradas ou desregistradas para monitorar o rastreamento da mão. Essas funções de retorno serão chamadas quando:
 - Uma nova mão for criada;
 - Uma mão existente trocas sua posição;
 - O rastreamento da mão for perdido (se a mão sumir)
- **UserGenerator:** Objeto que retorna dados relacionados ao ator em cena. Suas principais funcionalidades são:
 - **Get Number of Users:** fornece o número de usuários detectados em cena
 - **Get Users:** Fornece o usuário atual;
 - **Get User CoM:** Fornece a localização do centro de massa do usuário
 - **Get User Pixels:** Fornece os pixels que representam o usuário. A informação de saída é uma matriz de pixels da cena (*map of pixels*), onde os pixels que representam o usuário são rotulados com sua identificação (*User ID*)
 - **Register/Unregister user callbacks:** Habilita ou desabilita o registro das funções de retorno. Estas funções são chamadas quando um novo usuário for identificado ou quando este desaparecer da cena.

2.5.2.Criando um Projeto Vazio usando o OpenNI

Esta parte do trabalho descreve como configurar o ambiente de desenvolvimento para se construir aplicações usando o framework *OpenNI*, utilizando a IDE Eclipse CDT e o sistema operacional *Linux Ubuntu*.

Para se iniciar o desenvolvimento de uma aplicação com o *framework OpenNI* iremos criar um projeto novo clicando na aba *File*, depois em *New*, e logo após em *C++ Project*, como ilustra a Figura 1.12.

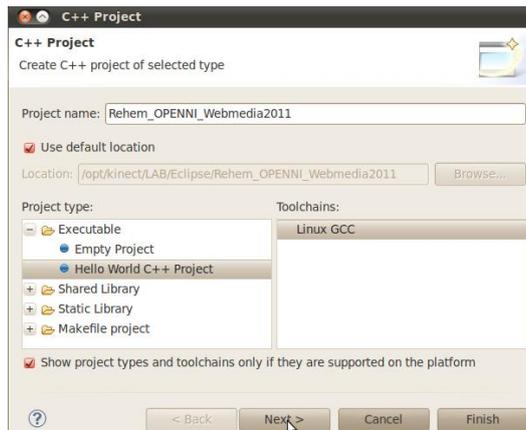


Figura 1.12: Criando um projeto do zero.

Em seguida, é necessário configurar as propriedades do projeto para que seja encontrado o caminho dos includes (arquivos .h) e das bibliotecas do *framework OpenNI* (arquivos .so), conforme ilustram as Figuras 1.Figura 1.13 e 1.Figura 1.14, respectivamente.

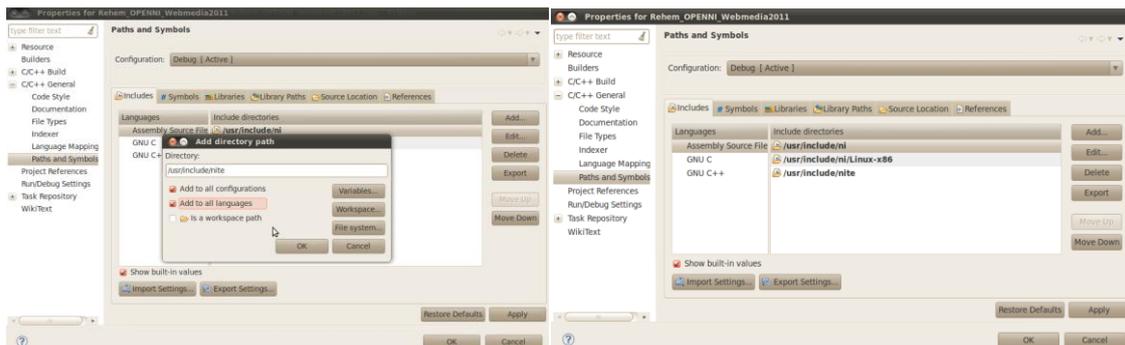


Figura 1.13. Configuração dos *includes* necessários para o *OpenNI*.

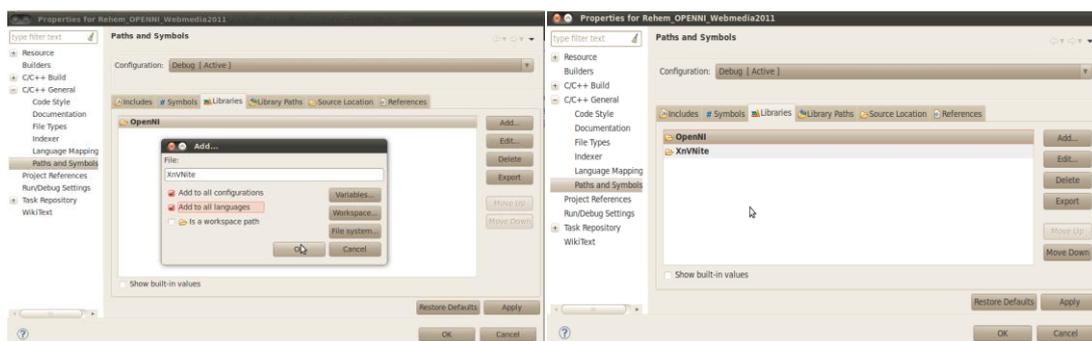


Figura 1.14. Configuração das bibliotecas necessárias o *OpenNI*.

É importante se certificar que os *Includes* e *Library directories* foram adicionados para as configurações de *Release* e de *Debug*.

Para iniciar a programação é necessário, que sejam incluídas as bibliotecas *XnOpenNI.h*, se for utilizar C, ou *XnCppWrapper.h*, se for utilizar C++ como linguagem de programação. A Figura a seguir ilustra um arquivo pronto para executar aplicações com objetos definidos pelo *framework OpenNI*.

```

Rehem_OPENNI_Webmedia2011.cpp  main.cpp
//=====
// Name       : Rehem_OPENNI_Webmedia2011.cpp
// Author      : Almerindo Rehem
// Version     :
// Copyright   : Touching The Air
// Description : Hello World in C++, Ansi-style
//=====

#include <XnCppWrapper.h>
using namespace xn;

#include <iostream>
using namespace std;

int main() {
    //TODO COLOCAR O CODIGO AQUI
    cout << "!!!Hello World!!!" << endl; // prints !!!Hello World!!!
    return 0;
}

```

Figura 1.15: Código com include para o C++ (XnCppWrapper.h).

2.5.2.1. Funções Básicas: Iniciando, Criando Nós de Produção e Lendo Dados

Como visto na seção 2.5.1. Principais Objetos do Framework, a classe *Context* é o objeto principal de uma aplicação que usa o *OpenNI*. É a partir dela que são acessados todos os outros objetos, tais como os nós de produção do tipo profundidade, RGB, infravermelho, áudio, dentre outros. Os trechos de códigos que se segue mostram como usar o *framework OpenNI* através da instância da classe *Context*, além de criar e ler dados de um nó de produção (*DepthGenerator*).

```

1  XnStatus nRetVal = XN_STATUS_OK;
2
3  xn::Context context;
4
5  // Initialize context object
6  nRetVal = context.Init();
7  // TODO: check error code

```

Trecho de Código 1.1. Instanciando a classe *Context*.

No código acima não existem nós de produção, ou seja, o contexto está vazio. Para se adicionar nós de produção à classe *Context*, é necessário criar esses nós dentro desta classe. O Trecho de Código 1.2. Criando um *Depth Generator* dentro de um *Context* mostra como criar um nó de produção do tipo *DepthGenerator* dentro de um *Context* já instanciado.

```

9  // Create a DepthGenerator node
10 xn::DepthGenerator depth;
11 nRetVal = depth.Create(context);
12 // TODO: check error code
13

```

Trecho de Código 1.2. Criando um *Depth Generator* dentro de um *Context*.

A partir do código acima, os nós de produção já estão plugados em um contexto, entretanto ainda não estão aptos a gerar conteúdo. Uma forma simples de fazer com que os *Generators* pertencentes a um *Context* gerem conteúdo é solicitar a este último

que inicie a produção de dados em todos os seus nós filhos. Este exemplo é descrito no Trecho de Código 1.3.

```
14 // Make it start generating data
15 nRetVal = context.StartGeneratingAll();
16 // TODO: check error code
17
```

Trecho de Código 1.3. Iniciando a produção de dados em todos os *Generators* filhos.

Com o objeto *Context* e seus nós de produção (neste caso, o *DepthGenerator*) instanciados e gerando conteúdos, se faz necessário capturar os dados todas as vezes que houver uma atualização. O método `context.WaitOneUpdateAll(depth)`, faz com que a classe *Context* espere por dados de um nó (*DepthGenerator*) e atualize todos os outros nós filhos. Com isso, basta capturar os dados (usando o `depth.GetDepthMap()`) e processá-los em seguida, conforme ilustrado no Trecho de Código 1.4.

```
18 // Main loop
19 while (bShouldRun)
20 {
21     // Wait for new data to be available
22     nRetVal = context.WaitOneUpdateAll(depth);
23     if (nRetVal != XN_STATUS_OK)
24     {
25         printf("Failed updating data: %s\n",
26 xnGetStatusString(nRetVal));
27         continue;
28     }
29
30     // Take current depth map
31     const XnDepthPixel* pDepthMap = depth.GetDepthMap();
32
33     // TODO: process depth map
34 }
```

Trecho de Código 1.4. Esperando por atualização e capturando dados.

Para liberar toda a memória utilizada nos trechos de códigos anteriores, é necessário chamar o método `Shutdown()` da classe *Context* conforme o código abaixo.

```
35
36 // Clean-up
37 context.Shutdown();
```

Trecho de Código 1.5. Liberando a memória utilizada.

2.5.2.2. A extração de esqueletos

O exemplo ilustrado nesta subseção mostra como detectar um usuário, monitora uma determinada pose do usuário, executa uma calibração quando este estiver fazendo uma pose específica e após calibração, segue a trajetória do usuário. Como saída, o sistema imprime em tela a localização da cabeça do usuário na cena. A Figura 1.16 ilustra as fases mais relevantes deste exemplo e permite um maior entendimento dos códigos descritos a seguir.



Figura 1.16. Funcionamento do exemplo de extração de esqueleto.

Para que seja possível executar as fases ilustradas na figura acima, é necessário se utilizar os recursos oferecidos pelo *framework* para monitorar os usuários, as poses e a calibração. A seguir, estes recursos serão discutidos, assim como sua utilização, através de alguns exemplos de códigos.

A classe *UserGenerator* oferece a possibilidade se registrar funções que são chamadas pelo próprio *framework* quando: (i) um novo usuário for detectado; (ii) o usuário corrente for perdido; (iii) uma pose for detectada; (iv) quando a calibração for iniciada e (v) quando a calibração for finalizada. O Trecho de Código 1.6 ilustra como criar as funções de retorno que alertam à aplicação quando um usuário for detectado e quando este sair de cena.

```

1  #define POSE_TO_USE "Psi"
2  xn::UserGenerator g_UserGenerator;
3
4  void XN_CALLBACK_TYPE
5  User_NewUser(xn::UserGenerator& generator, XnUserID nId, void* pCookie){
6      printf("New User: %d\n", nId);
7      g_UserGenerator.GetPoseDetectionCap().StartPoseDetection(POSE_TO_USE,nId);
8  }
9
10 void XN_CALLBACK_TYPE
11 User_LostUser(xn::UserGenerator& generator, XnUserID nId, void* pCookie){
12 }
13

```

Trecho de Código 1.6. Criando funções de retorno para detecção de usuários.

No Trecho de Código 1.7, a função `Pose_Detected()` será chamada quando uma pose for detectada. Ela irá imprimir na tela, o nome da pose e o identificador do usuário. Porém, é necessário desligar a detecção de pose (linha 17) e solicitar que seja feita a calibração do usuário (linha 18) para que seja possível extrair o seu esqueleto.

```

14 void XN_CALLBACK_TYPE
15 Pose_Detected(xn::PoseDetectionCapability& pose, const XnChar* strPose, XnUserID nId, void* pCookie){
16     printf("Pose %s for user %d\n", strPose, nId);
17     g_UserGenerator.GetPoseDetectionCap().StopPoseDetection(nId);
18     g_UserGenerator.GetSkeletonCap().RequestCalibration(nId, TRUE);
19 }

```

Trecho de Código 1.7. Criando função de retorno para detecção de pose.

A linha 18 do código acima solicita a calibração do usuário. Entretanto é necessário se saber quando a calibração foi iniciada, quando ela foi finalizada e se houve sucesso após seu término. Estas informações são obtidas por meio da implementação das funções `Calibration_Start()` e `Calibration_End()`, ilustradas no Trecho de Código 1.8 a seguir.

```

21 void XN_CALLBACK_TYPE
22 Calibration_Start(xn::SkeletonCapability& capability, XnUserID nId, void* pCookie)
23 {
24     printf("Starting calibration for user %d\n", nId);
25 }
26
27 void XN_CALLBACK_TYPE
28 Calibration_End(xn::SkeletonCapability& capability, XnUserID nId, XnBool bSuccess, void* pCookie){
29     if (bSuccess) {
30         printf("User calibrated\n");
31         g_UserGenerator.GetSkeletonCap().StartTracking(nId);
32     } else {
33         printf("Failed to calibrate user %d\n", nId);
34         g_UserGenerator.GetPoseDetectionCap().StartPoseDetection( POSE_TO_USE, nId);
35     }
36 }
37

```

Trecho de Código 1.8. Criando funções de retorno para calibração.

A função `Calibration_End()` (linha 28), verifica se a calibração do usuário obteve êxito, exibe em tela uma mensagem indicando que a calibração foi feita e chama o método `GetSkeletonCap().StartTracking()` (linhas 29-31), responsável por atualizar o esqueleto do usuário conforme sua posição em cena. Por outro lado, se não houver sucesso na calibração, será exibida uma mensagem de erro e retornará a detectar uma pose para possibilitar uma nova calibração (linhas 32-34).

Com as funções de retorno implementadas, é necessário que o objeto *UserGenerator* as reconheça no seu sistema de alerta. Para isso, as funções de retorno devem ser registradas através dos métodos `RegisterUserCallbacks()`, `RegisterToPoseCallbacks()`, `RegisterCalibrationCallbacks()`, conforme o código abaixo (linhas 50-53).

```

38 void main()
39 {
40     XnStatus nRetVal = XN_STATUS_OK;
41
42     xn::Context context;
43     nRetVal = context.Init();
44     // TODO: check error code
45
46     // Create the user generator
47     nRetVal = g_UserGenerator.Create(context);
48     // TODO: check error code
49
50     XnCallbackHandle h1, h2, h3;
51     g_UserGenerator.RegisterUserCallbacks(User_NewUser, User_LostUser, NULL, h1);
52     g_UserGenerator.GetPoseDetectionCap().RegisterToPoseCallbacks( Pose_Detected, NULL, NULL, h2);
53     g_UserGenerator.GetSkeletonCap().RegisterCalibrationCallbacks( Calibration_Start, Calibration_End, NULL, h3);
54
55     // Set the profile
56     g_UserGenerator.GetSkeletonCap().SetSkeletonProfile( XN_SKEL_PROFILE_ALL);
57     // Start generating
58     nRetVal = context.StartGeneratingAll();
59     // TODO: check error code

```

Trecho de Código 1.9. Registrando as funções de retorno no *UserGenerator*.

No exemplo a seguir, a quantidade máxima de usuários foi definida como sendo igual a quinze (linha 68). A partir deste ponto, todos os usuários em cena serão capturados por meio da função `GetUsers()` (linha 69). Em seguida, para cada um dos usuários que já estão sendo seguidos em cena (linhas 70 e 71), será extraída a posição do osso da cabeça deste usuário (linhas 72 e 73), sendo exibidas a sua identificação e a posição de sua cabeça (linhas 74-76).

```

60
61 while (TRUE)
62 {
63     // Update to next frame
64     nRetVal = context.WaitAndUpdateAll();
65     // TODO: check error code
66     // Extract head position of each tracked user
67     XnUserID aUsers[15];
68     XnUInt16 nUsers = 15;
69     g_UserGenerator.GetUsers(aUsers, nUsers);
70     for (int i = 0; i < nUsers; ++i){
71         if (g_UserGenerator.GetSkeletonCap().IsTracking(aUsers[i])){
72             XnSkeletonJointPosition Head;
73             g_UserGenerator.GetSkeletonCap().GetSkeletonJointPosition( aUsers[i], XN_SKEL_HEAD, Head);
74             printf("%d: (%f,%f,%f) [%f]\n", aUsers[i],
75                 Head.position.X, Head.position.Y,
76                 Head.position.Z, Head.fConfidence);
77         }
78     }
79 }
80

```

Trecho de Código 1.10. Capturando a cabeça do usuário e exibindo sua posição.

Por fim, é necessário liberar toda a memória utilizada pelo programa com o uso da função `Shutdown()`, conforme o Trecho de Código 1.11.

```

80
81 // Clean up
82 context.Shutdown();
83

```

Trecho de Código 1.11. Liberação da memória antes de sair do programa.

2.6. Conclusão

Este capítulo tratou do desenvolvimento de aplicações voltadas para interação natural utilizando o *framework OpenNI* e o *middleware NITE*. A intenção não foi esgotar o assunto, mas levantar questões importantes e divulgar as tecnologias associadas à Interação Natural. Foram apresentados conceitos de Interação Natural, comunicação verbal e não verbal, dispositivos envolvidos e sua evolução, assim como uma explicação mais detalhada sobre o *Kinect* e toda a motivação atual para as pesquisas relacionadas a este dispositivo. As seções finais do capítulo foram dedicadas ao *framework OpenNI* e à sua aplicação em um estudo de caso que mostrou o desenvolvimento, passo a passo, de aplicações em C++ para interação natural.

Como foi observado no capítulo, o desenvolvimento de aplicações para interação natural traz consigo vários desafios e oportunidades para os desenvolvedores de software e produtores de dispositivos de hardwares. Atualmente, nota-se uma tendência de utilização da plataforma *Kinect* não apenas na sua função original de controle de jogos, mas também como um novo periférico de entrada para comunicação usuário computador. Por outro lado, a área de IUC passa por um momento em que as metáforas de usabilidade estão se atualizando, e a interação natural usuário computador recebe um reforço importante na busca da ubiquidade computacional, simplificando a forma de fornecer entradas de dados para software e utilizar dispositivos tecnológicos em geral.

Desta maneira, o *OpenNI* está se tornando a principal aposta dos desenvolvedores de aplicações e comunidades científicas de interação natural usuário computador. Não só por ser um excelente *framework*, mas por ter se tornado uma plataforma padrão (*de facto*) de desenvolvimento para aplicações de interação natural. O projeto *OpenNI* abordado nesse capítulo oferece uma padronização de desenvolvimento, uma abstração maior dos elementos de hardware e uma facilidade de integração de outros componentes de softwares para interpretação dessas informações. Isto que permite tanto reuso das implementações feitas, quanto a leitura conjunta de sensores de mesma natureza aumentando a acuidade da informação.

2.7. Agradecimentos

Os autores agradecem à CAPES pelo suporte financeiro ao primeiro autor através da bolsa do Programa de Demanda Social para o Programa Multiinstitucional de Pós-graduação em Ciência da Computação UFBA/UNIFACS/UEFS (28001010061P1).

2.8. Referências

Abowd G.D, Mynatt E. D. (2000) “Charting past, present, and future research in ubiquitous computing.”ACM Transactions on Computer-Human Interaction. v.7, n.1, Mar. 2000.

Adafruit Industries (2011), “Open Kinect Challenge”. Disponível em URL: <http://www.adafruit.com/blog/2010/11/04/the-open-Kinect-project-the-ok-prize-get-1000-bounty-for-Kinect-for-xbox-360-open-source-drivers/>

- Cordeiro Jr. A. A. A. (2009) “Modelos e Métodos para Interação Homem-Computador Usando Gestos Manuais”. *Tese de Doutorado*. LNCC, Nov. 2009. URL: http://www.lncc.br/tcmc/tde_busca/arquivo.php?codArquivo=195
- Eclipse CDT Project (2011) “Eclipse CDT (C/C++ Development Tooling)”. URL: <http://www.eclipse.org/cdt/>
- Gallud, J.A.; Villanueva, P. G.; Tesoriero, R.; Sebastián, G.; Molina S.; A. Navarrete (2010). “Gesture-based Interaction. Concept Map and Application Scenarios”. DOI: *10.1109/CENTRIC.2010.10*
- Heckel, P. (1993) “Software amigável. Técnicas de projeto de software para uma melhor interface com o usuário”. Ed. Campus Ltda.
- Crawford, S. (2010) “How Microsoft Kinect Works”. 13/07/10. HowStuffWorks.com. URL: <http://electronics.howstuffworks.com/microsoft-kinect.htm>
- Kinect (2010) “Kinect Web Site”. Microsoft. Disponível em URL: <http://www.xbox.com/pt-br/Kinect>
- Kinect SDK (2010) “Kinect for Windows SDK”. Microsoft Research. Disponível em URL: <http://research.Microsoft.com/en-us/um/redmond/projects/Kinectsdk/default.aspx>
- Kinecthacks (2011). “Hacks para o Kinect”. Disponível em URL: <http://Kinecthacks.net/>.
- Kyma (2010). “Using Nitendo Wiimote With Kyma”. Kima Twiky. Disponível em URL: <http://www.symbolicsound.com/cgi-bin/bin/view/Learn/UsingTheNintendoWiimoteWithKyma>
- Nakra T., Ivanov Y., Smaragdis P., Ault C. (2009) “The USB Virtual Maestro: an Interactive Conducting System”, pp.250-255, *In: NIME2009*. Disponível em URL: www.cs.illinois.edu/~paris/pubs/nakra-nime09.pdf
- OpenNI (2011) “Introducing OpenNI”. OpenNI Organization. Disponível em URL: <http://www.OpenNI.org>
- OpenNI User Guide (2011). Disponível em URL: www.openni.org/images/stories/pdf/OpenNI_UserGuide.pdf
- Power Glove (2008) “Games: Manual de Instruções Power Glove”, Nintendo. Disponível em URL: <http://blablagames.net/?p=1006>
- PrimeSense (2011). “PrimeSense Natural Interaction”. PrimeSense. Disponível em URL: <http://www.primesense.com/>
- Projeto OpenKinect (2011) “Open Kinect Roadmap”. OpenKinect Org. Disponível em URL: <http://openKinect.org/wiki/Roadmap>.
- Saffer (2009) “*Designing Gestural Interfaces*”. D. Saffer, Books, O’Reilly. 2009.
- Shape Quest in (2010) “Point Cloud Data from Structured Light Scanning”. Disponível em URL: <http://www.shapecapture.com/gallery3.htm>
- Shiratori T.; Hodgins J. K. (2008) “Accelerometer-based user interfaces for the control of a physically simulated character”. *ACM Trans. Graph.* 27, 5, Article 123, Disponível em URL: <http://doi.acm.org/10.1145/1409060.1409076>.

- Toyama K. (1998) “Look, Ma – No Hands! Hands-Free Cursor Control with Real-Time 3D Face Tracking”. *URL:* <http://www.cs.ucsb.edu/conferences/PUI/PUIWorkshop98/Papers/Toyama.pdf>
- Valli A. (2007) “Natural Interaction White Paper”, Disponível em *URL:* <http://www.naturalinteraction.org/>
- Weiser M.; Brown J. S. (1995) “Designing calm technology”. Xerox PARC.
- Wii Remote (2009) “Wii Console Controllers”. Nintendo. Disponível em *URL:* <http://www.nintendo.com/wii/console/controllers>
- Wiimote (2010) “Wiiremote Commander”. Disponível em *URL:* <http://wiimotecommande.sourceforge.net/wiimote/index.html>

Capítulo

3

Linked Data: Construindo um Espaço de Dados Global na Web

Regis Pires Magalhães, José Antônio F. de Macêdo, Vânia Maria Ponte Vidal

Departamento de Computação, Universidade Federal do Ceará, Fortaleza-CE, Brasil
{regispires, vvidal, jose.macedo}@lia.ufc.br

Abstract

Currently the Web is not only a global space of linked documents. It is also becoming a huge global data space consisting of billions of RDF triples from many different domains, which is called Web of Data. Linked Data defines a set of principles that form the basis for the dissemination and use of the data on the Web. Since 2007 datasets from various domains have been published according to these principles, generating a growing volume of data and hence a demand for their consumption. This chapter provides a conceptual and practical base related to Linked Data, where its foundations are presented and tools for publishing and consuming such data are discussed. The chapter also presents applications that benefit from using data published according to Linked Data principles. It also addresses the state of the art in the area and discuss limitations, open questions and challenges to be overcome in the research domain about Linked Data.

Resumo

A Web atual deixou de ser apenas um espaço global de documentos interligados e está se tornando também um enorme espaço global de dados vinculados constituído de bilhões de triplas RDF que cobrem os mais variados domínios, denominada Web de Dados. Linked Data define um conjunto de princípios que formam a base para a difusão e uso de dados na Web. Desde 2007 conjuntos de dados dos mais diversos domínios têm sido publicadas de acordo com estes princípios, gerando um volume crescente de dados e, conseqüentemente, uma demanda por seu consumo. Este capítulo provê uma base conceitual e prática relacionada à Linked Data, onde são apresentados os seus fundamentos

e discutidas ferramentas para publicação e consumo de tais dados necessários para o desenvolvimento de aplicações. Apresenta ainda aplicações que se beneficiam do uso de dados publicados de acordo com os princípios *Linked Data*, além de tratar do estado da arte na área e discutir limitações, questões em aberto e desafios a serem superados no domínio da pesquisa sobre *Linked Data*.

3.1. Introdução

A Web atual deixou de ser apenas um espaço global de documentos interligados e está se tornando um enorme espaço global de dados vinculados constituído de bilhões de triplas RDF que cobrem os mais variados domínios [Heath and Bizer 2011]. Esta nova Web, denominada Web de Dados, visa pavimentar o caminho para a Web Semântica funcional, onde haverá a disponibilidade de uma grande quantidade de dados vinculados em formato RDF. Sua implementação é baseada nos princípios *Linked Data* delineados pelo diretor geral do W3C, o pesquisador Tim Berners-Lee. De fato, *Linked Data* é um conjunto de melhores práticas para publicação e conexão de dados estruturados na Web.

Inúmeras iniciativas voltadas para fomentar a criação da Web de Dados surgiram nos últimos anos, como por exemplo, o projeto *Linking Open Data (LOD)*¹ que é um esforço comunitário iniciado em 2007 e suportado pelo W3C para identificar fontes de dados publicadas sob licenças abertas, convertê-las para RDF e publicá-las na Web usando os princípios de *Linked Data*. Em outubro de 2010, este projeto havia publicado 207 conjuntos de dados compostos de mais de 28 bilhões de triplas RDF e aproximadamente 395 milhões de *links* RDF englobando os mais variados domínios como informações geográficas, censo, pessoas, empresas, comunidades online, publicações científicas, filmes, músicas, livros, além de outros [Bizer et al. 2011]. A figura 3.1 mostra um diagrama de nuvem com as fontes de dados publicadas pelo projeto LOD e as interligações entre elas em setembro de 2010. O tamanho dos círculos corresponde ao número de triplas de cada fonte de dados. As setas indicam a existência de pelo menos 50 links entre duas fontes. A origem de uma seta indica a fonte que possui o link e a fonte referenciada é a fonte para a qual a seta está apontando. Setas bidirecionais representam fontes que se referenciam mutuamente. A espessura da seta corresponde ao número de links.

Outra importante iniciativa foi a criação do *Workshop Linked Data on the Web (LDOW)*² dentro da programação da *International World Wide Web Conference (WWW2008)*, tendo, desde então, entre os membros de seu comitê organizador o pesquisador Tim Berners-Lee (W3C/MIT). No Brasil foi estabelecida no âmbito governamental a Infraestrutura Nacional de Dados Abertos (INDA)³, uma importante iniciativa criada em 2011 para aplicar os princípios de *Linked Data* na publicação de dados governamentais abertos.

A Web de Dados cria inúmeras oportunidades para a integração semântica de dados, fomentando o desenvolvimento de novos tipos de aplicações e ferramentas. Muito esforço tem sido despendido pela comunidade para o desenvolvimento de navegadores, mecanismos de busca e outras ferramentas específicas para consumo de dados vincula-

¹<http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

²<http://events.linkeddata.org/ldow2008/>

³<http://wiki.gtinda.ibge.gov.br/>

de *Linked Data* que são enumerados a seguir:

1. Usar URIs como nomes para coisas.
2. Usar URIs HTTP para que as pessoas possam procurar esses nomes.
3. Quando alguém procurar uma URI, prover informação útil, usando os padrões (RDF, SPARQL).
4. Incluir links para outras URIs, de modo que possam permitir a descoberta de mais coisas.

Esses princípios fornecem a base para a publicação e interligação de dados estruturados na Web. Posteriormente, eles foram estendidos por documentos originados a partir das experiências da comunidade de *Linked Data* [Bizer et al. 2007b, Sauermaun and Cyganiak 2008], resultando em boas práticas de publicação e consumo de *Linked Data* que serão apresentadas ao longo deste capítulo.

Para facilitar o entendimento da Web de dados, podemos estabelecer um paralelo com a Web de documentos que já conhecemos. A Web de dados pode ser acessada a partir de navegadores RDF, assim como os navegadores HTML são usados para acessar a Web de documentos. Enquanto na Web de documentos usamos links HTML para navegar entre diferentes páginas, na Web de dados os links RDF são usados para acessar dados de outras fontes. Portanto, os links de hipertexto são capazes de conectar os documentos, assim como os links RDF interligam os dados.

Além disso, a Web de documentos está alicerçada em um pequeno conjunto de padrões: um mecanismo de identificação global e único (URIs - Uniform Resource Identifiers), um mecanismo de acesso universal (HTTP - Hypertext Transfer Protocol) e um formato de conteúdo amplamente usado (HTML - Hypertext Markup Language). De modo semelhante, a Web de dados também tem por base alguns padrões bem estabelecidos como: o mesmo mecanismo de identificação usado na Web de documentos (URIs), um modelo de dados comum (RDF) e uma linguagem de consulta para acesso aos dados (SPARQL). O modelo RDF [Manola and Miller 2004] é baseado na idéia de identificar os recursos da Web usando identificadores (chamados Uniform Resource Identifiers - URIs⁴), e descrever tais recursos em termo de propriedades, as quais podem apontar para outras URIs ou ser representadas por literais. Esses padrões serão abordados a seguir:

URIs – Uniform Resource Identifiers

URIs [Berners-Lee et al. 2005] são usadas no contexto de *Linked Data* para identificar objetos e conceitos, permitindo que eles sejam dereferenciados para obtenção de informações a seu respeito. Assim, uma URI dereferenciada resulta em uma descrição RDF do recurso identificado. Por exemplo, a URI <http://www.w3.org/People/Berners-Lee/card#i> identifica o pesquisador Tim Bernes-Lee.

RDF – Resource Description Framework

A utilização um modelo de dados comum – modelo RDF – torna possível a implementação de aplicações genéricas capazes de operar sobre o espaço de dados global

⁴<http://www.ietf.org/rfc/rfc2396.txt>

[Heath and Bizer 2011]. O modelo RDF [Manola and Miller 2004] é um modelo de dados descentralizado, baseado em grafo e extensível, possuindo um alto nível de expressividade e permitindo a interligação entre dados de diferentes fontes. Ele foi projetado para a representação integrada de informações originárias de múltiplas fontes. Os dados são descritos na forma de triplas com sujeito, predicado e objeto, onde o sujeito é uma URI, o objeto pode ser uma URI ou um literal e o predicado é uma URI que define como sujeito e predicado estão relacionados. Por exemplo a afirmação em português '*http://www.w3.org/People/Berners-Lee/card#i tem uma propriedade denominada creator cujo valor é Tim Bernes-Lee*' pode ser definida através de uma tripla RDF da seguinte forma:

```
Sujeito: http://www.w3.org/People/Berners-Lee/card#i
Predicado: 'creator'
Objeto: 'Tim Bernes-Lee'
```

Cada tripla faz parte da Web de Dados e pode ser usada como ponto de partida para explorar esse espaço de dados. Tripas de diferentes fontes podem ser facilmente combinadas para formar um único grafo. Além disso, é possível usar termos de diferentes vocabulários para representar os dados. O modelo RDF ainda permite a representação de dados em diferentes níveis de estruturação, sendo possível representar desde dados semi-estruturados a dados altamente estruturados.

RDF Links

No contexto de *Linked Data* os *RDF links* descrevem relacionamentos entre dois recursos [Heath and Bizer 2011]. Um *RDF link* consiste de três referências URI. As URIs referentes ao sujeito e objeto identificam os recursos relacionados. A URI referente ao predicado define o tipo de relacionamento entre os recursos. Um distincao util que pode ser feita é com relação à links internos e externos. *RDF links* internos conectam recursos dentro de uma unica fonte de dados Linked Data. Links externos conectam recursos qua sao servidos por diferentes fontes de dados Linked Data. No caso de links externos, as URIs referentes ao sujeito e predicado do link pertencem a diferentes namespaces. Links externos são cruciais para a Web dos Dados visto que eles permitem juntar as fontes de dados dispersas em um espaço global de dados.

A figura 3.2 apresenta dois exemplos de *RDF links*. O primeiro exemplo interliga o perfil FOAF do pesquisador Tim Berners-Lee localizado em um arquivo RDF ao recurso que o identifica na fonte de dados do DBLP. No segundo exemplo, o recurso que identifica Tim Berners-Lee na fonte DBpedia também é ligado ao recurso na fonte DBLP que o identifica. A propriedade *http://www.w3.org/2002/07/owl#sameAs* define que os recursos interligados representam a mesma entidade do mundo real.

O armazenamento de dados no modelo RDF pode ser realizado através de grafo em memória, arquivo texto ou banco de dados específico para armazenamento de triplas RDF, chamado de *RDF Triple Store*. O armazenamento de triplas em arquivo texto usa algum formato de serialização de RDF, como RDF/XML, Notation3 (N3), Turtle ou NTriples.

Protocolo e Linguagem SPARQL

Consultas à Web de Dados podem ser realizadas através da linguagem SPARQL

Sujeito: http://www.w3.org/People/Berners-Lee/card#i Predicado: http://www.w3.org/2002/07/owl#sameAs Objeto: http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007
Sujeito: http://dbpedia.org/resource/Tim_Berners-Lee Predicado: http://www.w3.org/2002/07/owl#sameAs Objeto: http://www4.wiwiss.fu-berlin.de/dblp/resource/person/100007

Figura 3.2. Exemplos de links RDF

[Prud'hommeaux and Seaborne 2008], que é a linguagem de consulta padrão da Web Semântica para recuperação de informações contidas em grafos RDF. No entanto, SPARQL não é somente uma linguagem de consulta declarativa, mas também um protocolo [Clark et al. 2008] usado para enviar consultas e recuperar resultados através do protocolo HTTP.

Fontes *Linked Data* tipicamente fornecem um *SPARQL endpoint* que é um serviço Web com suporte ao protocolo SPARQL. Esse serviço possui uma URI específica para receber requisições HTTP com consultas SPARQL e retornar os resultados dessas consultas. Os resultados podem ter diferentes formatos. Consultas que usam os comandos SELECT e ASK geralmente são retornadas nos formatos XML, JSON ou texto plano. Já os resultados de consultas através dos comandos DESCRIBE ou CONSTRUCT normalmente usam os formatos RDF/XML, NTriples, Turtle ou N3. Caso o *endpoint* não receba os parâmetros exigidos pelo protocolo SPARQL, ele exibe uma página HTML interativa que permite ao usuário digitar e submeter uma consulta.

Além do uso dos padrões acima descritos, a Web de dados possui as seguintes características de acordo com [Bizer et al. 2009]:

1. É genérica e pode conter qualquer tipo de dado;
2. qualquer pessoa pode publicar dados;
3. não há restrições para seleção de vocabulários;
4. os dados são auto-descritos, de modo que ao dereferenciá-los é possível obter sua definição;
5. o uso de um mecanismo padrão de acesso aos dados (HTTP) e de um modelo de dados padrão (RDF) simplifica o acesso aos dados;
6. as aplicações que usam a Web de dados não ficam limitadas a um conjunto fixo de fontes de dados, podendo inclusive descobrir novas fontes de dados em tempo de execução ao seguir links RDF.

3.3. Publicação de *Linked Data*

Publicar *Linked Data* significa usar os princípios e melhores práticas de *Linked Data* para disponibilizar os dados na Web. Basicamente isso quer dizer que é preciso fornecer URIs

dereferenciáveis para cada entidade e criar links RDF para outras fontes de dados. Esses são os requisitos mínimos, mas além deles é frequente a disponibilização de *SPARQL endpoints* e de *dumps* dos dados.

A publicação de *Linked Data* normalmente envolve a disponibilização de uma **interface *Linked Data*** capaz de tratar requisições de URIs, dereferenciar URIs, tratar dos redirecionamentos 303 requeridos pela arquitetura Web e da negociação de conteúdo entre descrições de um mesmo recurso em diferentes formatos. Através da negociação de conteúdo pode-se retornar a representação mais adequada ao cliente. Assim, um usuário humano pode requisitar uma URI e obter uma representação HTML do recurso. Isso é possível porque os clientes HTTP enviam cabeçalhos indicando que tipo de representação eles preferem obter. Daí o servidor analisa o cabeçalho ao receber uma requisição e seleciona a resposta adequada. Vejamos um exemplo disso ao acessar a URI http://dblp.l3s.de/d2r/resource/Marco_A._Casanova de uma fonte DBLP. Se o cabeçalho usado na requisição especificar o conteúdo `text/html`, o servidor enviará para o cliente um redirecionamento para a URI http://dblp.l3s.de/d2r/page/Marco_A._Casanova que possui como conteúdo uma página HTML. Caso seja usado um cabeçalho especificando conteúdo `application/rdf+xml`, o cliente receberá como resposta um redirecionamento para a URI http://dblp.l3s.de/d2r/data/Marco_A._Casanova que contém o resultado do dereferenciamento da URI em formato RDF/XML. Pubby⁵ é uma ferramenta para fornecer de forma simples uma interface *Linked Data* para fontes de dados RDF, como *SPARQL endpoints* e arquivos RDF estáticos.

Outro recurso frequentemente relacionado à publicação de *Linked Data* é o fornecimento de um *SPARQL endpoint* para possibilitar a realização de consultas SPARQL sobre uma fonte de dados. Alguns *endpoints* permitem inclusive atualizações através de *SPARQL Update* [Gearon et al. 2011]. O projeto Jena⁶ disponibiliza duas implementações de *SPARQL endpoints*: o *Joseki*⁷ e o *Fuseki*⁸. Ambos suportam o protocolo e a linguagem SPARQL; permitem a realização de consultas sobre arquivos RDF e *RDF Triple Stores*; e implementam SPARQL Update. A diferença principal entre eles é que o *Fuseki* é mais simples de configurar e usar, além de já possuir a *RDF Store Jena TDB*⁹ embutida. No entanto o *Fuseki* é um projeto mais recente iniciado em 2011 que ainda possui limitações quanto ao gerenciamento de múltiplas fontes de dados e também quanto ao uso de mecanismos de segurança próprios.

3.3.1. Publicação de dados de fontes RDF como *Linked Data*

As fontes de dados que adotam o modelo RDF são normalmente armazenadas em arquivo RDF ou *RDF Store*. A seleção da forma de armazenamento mais adequada vai depender de fatores como o volume de dados utilizado e a frequência de atualização desses dados. Nesta subseção veremos como publicar arquivos RDF ou *RDF Stores* como *Linked Data*.

Publicação de arquivos RDF como *Linked Data*

⁵<http://www4.wiwiw.fu-berlin.de/pubby/>

⁶<http://incubator.apache.org/jena/>

⁷<http://www.joseki.org/>

⁸<http://openjena.org/wiki/Fuseki>

⁹<http://openjena.org/TDB/>

Criar arquivos RDF estáticos e disponibilizá-los através de um servidor Web é a forma mais simples de publicar *Linked Data*. Essa estratégia somente é viável quando o arquivo RDF é relativamente pequeno. Embora haja vários formatos para serialização de um arquivo RDF (RDF/XML, Turtle, Notation3, NTriples), o formato mais antigo e também mais usado para publicação de *Linked Data* é o RDF/XML. Assim, se o arquivo for escrito manualmente em um formato mais simples, compacto e legível para o ser humano como Turtle, deverá ser posteriormente convertido para RDF/XML. A figura 3.3 apresenta um trecho de um arquivo RDF no formato Turtle que possui informações sobre organizações e pesquisadores. Cada pesquisador possui um *RDF link* para o recurso correspondente na fonte de dados DBLP. Interessante observar que o RDF mostrado na figura adota a boa prática de somente usar vocabulários existentes. Nesse caso, fizemos uso de termos dos vocabulários FOAF e DC.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://lia.ufc.br/~regispires/researchers.rdf#> .

# The <> (the empty URI) means "this document".
<> a foaf:Document ;
   dc:title "Researchers file" .

:ufc
  a foaf:Organization ;
  foaf:name "Universidade Federal do Ceará" .

:vania
  a foaf:Person ;
  foaf:name "Vânia Maria Ponte Vidal" ;
  foaf:Organization :ufc ;
  rdfs:sameAs <http://dblp.13s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal> .
```

Figura 3.3. Arquivo RDF serializado no formato Turtle

A conversão de Turtle para RDF/XML pode ser realizada através de algum conversor online como *Babel*¹⁰ ou *RDFConverter*¹¹. A figura 3.4 apresenta o arquivo RDF/XML a ser publicado através de um servidor Web. Os recursos presentes no arquivo RDF podem ser acessados através do uso de *Hash URIs* [Sauer mann and Cyganiak 2008] que possuem um identificador de fragmento adicionado ao nome do arquivo. Em nosso exemplo, o identificador de fragmento #vania é adicionado à URI do arquivo para que a URI <http://lia.ufc.br/regispires/researchers.rdf#vania> possa ser usada para acessar o recurso correspondente. O nome *Hash URI* deve-se ao uso do símbolo # que é representado pela palavra *hash* na língua inglesa.

A disponibilização de um arquivo RDF/XML em um servidor Web possibilita o dereferenciamento de URIs e o uso de *RDF links*, mas ainda não resolve os redirecionamentos 303 e a negociação de conteúdo. Uma interface *Linked Data* para solucionar essas questões pode ser criada ou mesmo provida pelo serviço Pubby¹² através do uso da opção de configuração *loadRDF* que possibilita o carregamento de um arquivo RDF estático para a memória e o fornecimento de uma interface *Linked Data* para ele.

¹⁰<http://simile.mit.edu/babel>

¹¹<http://www.mindswap.org/2002/rdfconvert/>

¹²<http://www4.wiwiss.fu-berlin.de/pubby/>

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns="http://lia.ufc.br/~regispires/researchers.rdf#">

  <rdf:Description rdf:about="">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
    <dc:title>Researchers file</dc:title>
  </rdf:Description>

  <rdf:Description rdf:about="http://lia.ufc.br/~regispires/researchers.rdf#ufc">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
    <foaf:name>Universidade Federal do Ceará</foaf:name>
  </rdf:Description>

  <rdf:Description rdf:about="http://lia.ufc.br/~regispires/researchers.rdf#vania">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:name>Vânia Maria Ponte Vidal</foaf:name>
    <foaf:Organization rdf:resource="http://lia.ufc.br/~regispires/researchers.rdf#ufc"/>
    <rdfs:sameAs
  rdf:resource="http://dblp.l3s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal"/>
  </rdf:Description>
</rdf:RDF>

```

Figura 3.4. Arquivo RDF serializado no formato RDF/XML

Dados RDF também poder vir embutidos em documentos Web. RDFa [Adida and Birbeck 2008] permite a inclusão de dados RDF em um documento XHTML, aproximando a Web de Documentos da Web de Dados (Figura 3.5 apresenta o conteúdo de um documento XHTML usando RDFa). Desse modo, textos e *links* legíveis ao ser humano também podem coexistir com dados relacionados que poderão ser facilmente processados pelas máquinas. As alterações ficam centralizadas em um único documento. Atualmente algumas ferramentas de publicação de conteúdo estão adicionando RDFa a seus documentos XHTML. Um exemplo disso é o Sistema de Gerenciamento de Conteúdo Drupal¹³ a partir de sua versão 7.

Até aqui falamos de arquivos RDF com conteúdo estático. No entanto, os arquivos RDF ou RDFa podem ter seus conteúdos gerados e preenchidos dinamicamente através de uma aplicação Web.

Publicação de dados de *RDF Store* como *Linked Data*

A publicação de dados de uma *RDF Store* como *Linked Data* tipicamente envolve a disponibilização de uma interface *Linked Data* e de um *SPARQL endpoint* para acesso aos dados. O ideal seria que toda *RDF Store* já tivesse esses recursos integrados para que somente fosse necessário especificar que dados seriam publicados como *Linked Data*. Entretanto em muitos casos há necessidade de se instalar serviços adicionais para prover tais recursos.

Se a *RDF Store* não fornecer um *SPARQL endpoint* embutido, este serviço pode ser instalado para permitir a execução de consultas SPARQL sobre a *RDF Store*. Um servidor *Joseki* ou *Fuseki* pode ser usado como *SPARQL endpoint* de uma *RDF Store*.

¹³<http://drupal.org/>

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/Markup/DTD/xhtml-rdfa-1.dtd">
<html
  xml:lang="en"
  version="XHTML+RDFa 1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rs="http://lia.ufc.br/~regispires/researchers.rdf#"
>
  <head>
    <title>Researchers</title>
    <meta http-equiv="content-type" content="text/html;charset=UTF-8" />
    <meta property="dc:title" content="Researchers file" />
    <link rel="rdf:type" href="http://xmlns.com/foaf/0.1/Document" />
  </head>
  <body>
    <h1>Organizations</h1>
    <ul>
      <li>
        <div about="http://lia.ufc.br/~regispires/researchers.rdf#ufc" typeof="foaf:Organization">
          <span property="foaf:name">Universidade Federal do Ceará</span>
        </div>
      </li>
    </ul>
    <h1>Researchers</h1>
    <ul>
      <li>
        <div about="http://lia.ufc.br/~regispires/researchers.rdf#vania" typeof="foaf:Person">
          <span property="foaf:name">Vânia Maria Ponte Vidal</span>
          <span rel="foaf:Organization" resource="http://lia.ufc.br/~regispires/researchers.rdf#ufc"></span>
          <span rel="rdfs:sameAs" resource="http://dblp.13s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal"></span>
        </div>
      </li>
    </ul>
  </body>
</html>

```

Figura 3.5. Exemplo de RDFa

O *Joseki* provê suporte nativo para acesso às *RDF Stores Jena TDB*¹⁴ e *Jena SDB*¹⁵. O *Fuseki* já inclui a *RDF Store Jena TDB*.

Caso a *RDF Store* não possua uma interface *Linked Data*, a aplicação *Pubby* pode ser usada para prover essa funcionalidade a partir de consultas realizadas sobre o *SPARQL endpoint*. Assim, ao receber uma requisição contendo uma URI a ser dereferenciada, o *Pubby* realiza uma consulta SPARQL usando o comando DESCRIBE para obtenção do resultado em formato RDF. Dependendo na negociação de conteúdo, esse resultado pode ser convertido para algum formato de serialização de RDF ou mesmo para a exibição de uma página HTML a ser exibida no navegador do usuário.

3.3.2. Publicação de dados de fontes não RDF como *Linked Data*

Como RDF é o modelo de dados comum usado em *Linked Data*, os dados publicados devem estar nesse modelo ou serem convertidos para ele. Caso os dados não adotem o modelo RDF, há duas abordagens possíveis para tratar essa heterogeneidade:

(i) Usar um processo de **conversão**, onde os dados não RDF são usados para gerar um arquivo RDF através de uma ferramenta específica¹⁶. Desse modo, através de conversores específicos é possível converter planilhas, arquivos CSV, arquivos XML e outros docu-

¹⁴<http://openjena.org/TDB/>

¹⁵<http://openjena.org/SDB/>

¹⁶<http://www.w3.org/wiki/ConverterToRdf>

mentos para o formato RDF. O projeto *RDFizer*¹⁷ contém informações de ferramentas para conversão de vários formatos de dados para RDF, além de hospedar algumas dessas ferramentas. Após geração do arquivo em formato RDF, seus dados podem ser carregados em uma *RDF Store* e publicados conforme descrito na subseção 3.3.1. Uma vantagem dessa abordagem é a melhoria de desempenho que pode ser obtida ao usar formas de armazenamento especificamente otimizadas para realizar a persistência de triplas RDF. No entanto, o armazenamento das triplas requer espaço extra em relação aos dados originais. Além disso, a conversão demanda um certo tempo para ser realizada e os dados em RDF podem ficar desatualizados em relação aos dados originais.

(ii) Fornecer uma **visão RDF** para acesso a dados que não estão no modelo RDF através de um *RDF Wrapper*. A conversão dinâmica realizada pelo *RDF Wrapper* baseia-se em mapeamentos estabelecidos entre o modelo nativo e o modelo RDF, devendo haver um *Wrapper* específico para cada tipo de modelo. Um *RDF Wrapper* também pode prover uma visão RDF a dados que precisam ser acessados através de uma Web API. *RDF Book Mashup* [Bizer et al. 2007a] é uma aplicação *mashup* escrita em PHP que funciona como um *RDF Wrapper* usado para combinar dados obtidos a partir das APIs proprietárias *Amazon Web Service* e *Google Base*. Desse modo, informações sobre livros, autores, revisões e comparações de ofertas entre diferentes livrarias podem ser usados por clientes genéricos, incluindo navegadores e mecanismos de busca RDF. Essa abordagem tende a ter um desempenho inferior à abordagem anterior (i) devido às traduções dinâmicas entre os modelos que deve ser realizada a cada uso da visão RDF e também devido ao modelo original não estar otimizado especificamente para uso de triplas. No entanto, o uso de *RDF Wrappers* traz grandes vantagens, pois como o acesso ocorre sobre os dados originais, a visão RDF não requer espaço de armazenamento extra e não corre o risco de apresentar dados desatualizados.

A ampla difusão dos Bancos de Dados Relacionais motiva a necessidade de publicação dos dados no modelo relacional como *Linked Data*. Assim, seguindo a abordagem (ii), há *RDB-to-RDF Wrappers* que criam visões RDF a partir de mapeamentos entre as estruturas relacionais e os grafos RDF. A plataforma D2RQ¹⁸ [Bizer and Seaborne 2004] fornece a infra-estrutura necessária para acessar bancos de dados relacionais como grafos RDF virtuais. Ela possui os seguintes componentes:

- **Linguagem de mapeamento D2RQ** é uma linguagem declarativa para descrever as correspondências entre o modelo relacional e o modelo RDF. Os mapeamentos escritos em D2RQ são documentos RDF.
- **Mecanismo D2RQ** é um plug-in para os *frameworks Jena* e *Sesame* que usa os mapeamentos escritos na linguagem D2RQ para converter chamadas às APIs desses *frameworks* em consultas SQL ao banco de dados para obtenção dos resultados.
- **Servidor D2R**¹⁹ [Bizer and Cyganiak 2006] é um servidor HTTP que usa o mecanismo D2RQ para prover uma interface *Linked Data* e um *SPARQL endpoint* sobre o banco de dados relacional.

¹⁷<http://simile.mit.edu/RDFizers/>

¹⁸<http://www4.wiwiss.fu-berlin.de/bizer/d2rq/spec/>

¹⁹<http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/>

Os componentes anteriormente descritos funcionam de forma integrada como pode ser observado na figura 3.6 que apresenta a arquitetura da plataforma D2RQ.

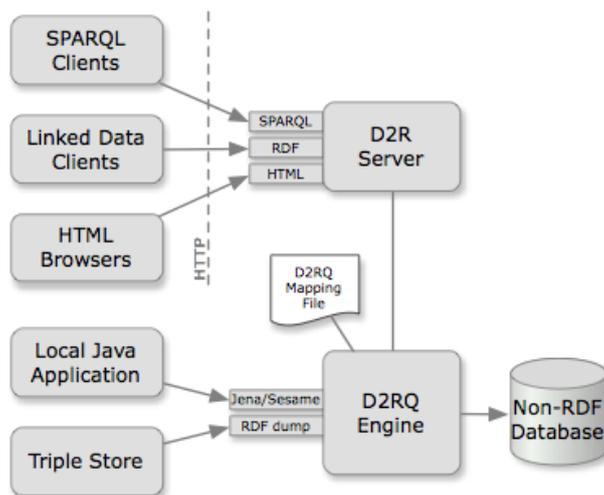


Figura 3.6. Arquitetura da plataforma D2RQ extraída de <http://www4.wiwiwss.fu-berlin.de/bizer/d2rq/spec/>

Além do D2R, duas outras ferramentas se destacam como *RDB-to-RDF Wrappers*: o Virtuoso RDF Views [Erling and Mikhailov 2006] e o Triplify²⁰ [Auer et al. 2009]. Este último é um pequeno *plugin* para aplicações Web que permite mapear os resultados de consultas SQL em RDF, JSON e *Linked Data*. Depois disso, os dados podem ser compartilhados e acessados na web de dados. *Triplify* consiste de poucos arquivos, totalizando menos de 500 linhas de código.

Um importante passo para a padronização desse tipo de solução para lidar com o modelo relacional foi a criação em 2009 do grupo de trabalho RDB2RDF²¹ do W3C. Desde então, o grupo tem definido a linguagem padrão R2RML [Das et al. 2011] visando o mapeamento de dados e esquemas relacionais para RDF, que tende a substituir as soluções de mapeamento já existentes.

Processo de Triplificação de Modelo Relacional para RDF

O processo *StdTrip* [Salas et al. 2010] guia usuários durante os estágios de modelagem conceitual do processo RDB para RDF, que pode ser definido como uma tradução do modelo relacional para o modelo RDF. A maioria das ferramentas RDB para RDF atuais realizam essa tarefa através do mapeamento de tabelas para classes RDF e atributos para propriedades RDF, sem a preocupação de identificar possíveis correspondências com vocabulários padrões existentes. Em vez disso, essas ferramentas criam novos vocabulários. O processo *StdTrip* baseia-se nesse princípio e busca promover o reuso de padrões pela implementação de um processo guiado composto de seis passos. Os passos 1 a 6 são denominados respectivamente Conversão, Alinhamento, Seleção, Inclusão, Complementação e Saída, de acordo com as principais operações realizadas em cada um. Enquanto os passos 1, 2, 3 e 6 são obrigatórios, os passos 4 e 5 são opcionais. A seguir, uma descrição sucinta desses passos:

²⁰<http://triplify.org/>

²¹<http://www.w3.org/2001/sw/rdb2rdf/>

1. **Conversão.** Este passo consiste em transformar a estrutura do banco de dados relacional em uma ontologia RDF. O projetista pode usar abordagens como *W-Ray* [Piccinini et al. 2010], onde ele manualmente define um conjunto de visões do banco de dados que capturam os dados que devem ser publicados e, então, especifica modelos (*templates*) que indicam como as triplas RDF devem ser geradas.
2. **Alinhamento.** Este passo usa a ferramenta de alinhamento de ontologias *K-match*²² para obter correspondências entre a ontologia obtida no passo 1 e um conjunto de vocabulários padrões. O processo de alinhamento considera o esquema da ontologia previamente obtido como o esquema fonte que será recursivamente alinhado com cada ontologia representando vocabulários padrões (ontologias destino).
3. **Seleção.** Este passo apresenta ao usuário as listas de possibilidades a fim de que ele selecione os elementos dos vocabulários que melhor representam cada conceito no banco de dados. Desse modo, listas de possíveis correspondências são apresentadas para cada elemento do esquema (tabela ou atributo).
4. **Inclusão.** Se, para um dado elemento, o processo não dá um resultado (não há elemento nos vocabulários conhecidos que corresponda ao conceito no banco de dados) ou nenhuma das sugestões da lista é considerada adequada pelo usuário, *StdTrip* provê uma lista de possíveis correspondências de outros vocabulários. A escolha desses vocabulários é dependente do domínio e a busca baseada em palavras-chave é realizada através do mecanismo de busca *Watson* (ver seção 3.4).
5. **Complementação.** Se nada funciona, usuários são direcionados às melhores práticas de publicação de vocabulários RDF [Berrueta and Phipps 2008].
6. **Saída.** O processo resulta em dois artefatos: (1) um arquivo de configuração que serve como a parametrização de uma ferramenta RDB para RDF. (2) uma ontologia que contém os mapeamentos do esquema do banco de dados original para vocabulários RDF padrões.

3.3.3. Melhores práticas para publicação de *Linked Data*

A adoção das melhores práticas de publicação de *Linked Data* facilita a descoberta de informações relevantes para a integração de dados entre diferentes fontes. A seguir serão descritas algumas dessas práticas.

- **Selecionar URIs adequadas.** Deve-se evitar URIs contendo algum detalhe de implementação ou do ambiente em que estão publicadas. Como exemplo a evitar, consideremos o URI <http://lia.ufc.br:8080/regispres/cgi-bin/resource.php?id=ufc> que possui detalhes da porta 8080 usada em seu ambiente de publicação e do script implementado em PHP necessário à sua execução.

É frequente o uso de três URIs relacionadas a cada recurso: (i) um identificador para o recurso; (ii) um identificador para informações sobre o recurso para visualização através de navegadores HTML; (iii) um identificador para informações sobre

²²K-match combina os resultados de algumas ferramentas de correspondência para a obtenção de um resultado mais preciso.

o recurso em formato RDF/XML. A figura 3.7 representa um exemplo de três URIs relacionadas à pesquisadora Vânia Vidal na fonte DBLP. A representação de um recurso através diferentes URIs permite que a interface Linked Data realize o dereferenciamento da URI de acordo com o tipo de conteúdo requisitado no cabeçalho HTTP (i.e. Text/HTML, application/rdf+xml, etc).

```
http://dblp.13s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal
http://dblp.13s.de/d2r/page/V%C3%A2nia_Maria_Ponte_Vidal
http://dblp.13s.de/d2r/data/V%C3%A2nia_Maria_Ponte_Vidal
```

Figura 3.7. Exemplos de URIs relacionadas a um mesmo recurso

A figura 3.8 apresenta dois exemplos de requisições HTTP referente à URI da pesquisadora Vânia Vidal na fonte DBLP. No exemplo referente ao item (a) a requisição define como tipo MIME dados no modelo RDF e recebe como resposta (através do redirecionamento 303 a URI referente aos dados da pesquisadora), no exemplo referente ao item (b) a requisição solicita os dados no formato HTML e recebe como resposta o redirecionamento para a URI referente à pagina HTML da pesquisadora.

```
(a)
$ curl -H "Accept: application/rdf+xml"
  http://dblp.13s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal

303 See Other: For a description of this item,
see http://dblp.13s.de/d2r/data/V%C3%A2nia_Maria_Ponte_Vidal

(b)
$ curl -H "Accept: text/html"
  http://dblp.13s.de/d2r/resource/V%C3%A2nia_Maria_Ponte_Vidal

303 See Other: For a description of this item,
see http://dblp.13s.de/d2r/page/V%C3%A2nia_Maria_Ponte_Vidal
```

Figura 3.8. Exemplos de requisições HTTP com tipos MIME RDF e HTML

- **Usar URIs dereferenciáveis** para que a descrição do recurso possa ser obtida da Web.
- **Utilizar URIs estáveis.** A alteração de URIs quebra links já estabelecidos, criando um problema para a localização de recursos. Para evitar esse tipo de alteração, recomenda-se um planejamento metódico das URIs que serão usadas e também que o responsável pela publicação detenha a propriedade do espaço de nomes.
- **Criar links para outras fontes de dados** de modo a permitir a navegação entre as fontes de dados. Os *links* podem ser criados de forma manual ou automatizada.
- **Publicação de Metadados.** Análise dos metadados facilita a seleção dos dados relevantes. Devem ser fornecidos metadados sobre proveniência e licenciamento dos dados. Também é recomendável a disponibilização de metadados sobre a fonte de dados.

- **Usar termos de vocabulários amplamente usados.** Embora não haja restrições para seleção de vocabulários, é considerada uma boa prática o reuso de termos de vocabulários RDF amplamente usados para facilitar o processamento de *Linked Data* pelas aplicações clientes [Bizer et al. 2007b]. Novos termos só devem ser definidos se não forem encontrados em vocabulários já existentes. A seguir apresentamos alguns vocabulários bastante difundidos: *Friend-of-a-Friend* (FOAF), *Semantically-Interlinked Online Communities* (SIOC), *Simple Knowledge Organization System* (SKOS), *Description of a Project* (DOAP), *Creative Commons* (CC) e *Dublin Core* (DC). Uma relação mais extensa desses vocabulários é mantida pelo projeto *Linking Open Data* no *ESW Wiki*²³.
- **Estabelecer relações entre os termos de vocabulários proprietários para termos de outros vocabulários.** Isso pode ser feito através do uso das propriedades *owl:equivalentClass*, *owl:equivalentProperty*, *rdfs:subClassOf*, *rdfs:subPropertyOf*. A figura 3.9 mostra que a classe Pessoa de um vocabulário local é equivalente à definição da classe *Person* no vocabulário da *DBpedia*.

```
<http://lia.ufc.br/Pessoa> owl:equivalentClass <http://dbpedia.org/ontology/Person> .
```

Figura 3.9. Relação de equivalência entre termo proprietário e termo da DBpedia

- **Explicitar formas de acesso adicional aos dados** como *SPARQL endpoints* e *RDF dumps*.

3.3.4. Validação dos dados publicados como *Linked Data*

Não basta simplesmente publicar dados RDF para que eles sejam considerados *Linked Data*. É necessário garantir que eles realmente estão de acordo com essas melhores práticas. Isso pode ser realizado através das ferramentas de validação que serão apresentadas a seguir.

O *W3C Validation Service*²⁴ permite a validação de um arquivo RDF antes de sua publicação. Documentos RDFa podem ser validados através do *W3C RDFa Distiller and Parser*²⁵.

*Eyeball*²⁶ é uma ferramenta criada pelo projeto Jena para a checagem de problemas comuns em modelos RDF. Ele fornece uma série de mensagens auto-explicativas sobre problemas existentes no modelo RDF analisado.

cURL é uma ferramenta de linha de comando bastante útil para checagem de tipos de conteúdos usados, redirecionamentos 303 e negociação de conteúdo decorrentes de acessos à URIs [Cyganiak 2007]. Algumas extensões de navegadores Web para modificação de cabeçalhos HTTP também podem ser usadas para realizar esses mesmos tipos de validações providas pelo *cURL*.

²³<http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/CommonVocabularies>

²⁴<http://www.w3.org/RDF/Validator/>

²⁵<http://www.w3.org/2007/08/pyRdfa/>

²⁶<http://jena.sourceforge.net/Eyeball/>

Vapour [Berrueta et al. 2008] é um serviço online²⁷ de validação que a partir de uma URI, checa se os dados estão publicados de acordo com as melhores práticas definidas pelos documentos: Princípios de *Linked Data* [Berners-Lee 2006], *Best Practice Recipes* [Berrueta and Phipps 2008] e *Cool URIs* [Sauermann and Cyganiak 2008]. Ele fornece um relatório detalhado sobre os dereferenciamentos de URIs realizados por ele. Além de serviço online, *Vapour*²⁸ também é disponibilizado como software livre escrito em Python sob licença W3C.

*Sindice Web Data Inspector*²⁹ é uma ferramenta online para visualizar, inspecionar e validar o conteúdo de dados estruturados disponível em uma determinada URL ou através do preenchimento de uma caixa de texto. Esse conteúdo pode ser de arquivos RDF, páginas HTML com microformatos embutidos ou páginas XHTML contendo RDFa. No final do processo é apresentado um relatório completo com os resultados obtidos.

Os navegadores RDF abordados na seção 3.4 também são úteis para a detecção de problemas a partir da análise das visualizações de URIs dereferenciadas por eles.

3.4. Consumo de *Linked Data*

URIs, palavras-chave e consultas SPARQL são usados como ponto de partida para o consumo de *Linked Data*. Assim, todas as aplicações que consomem a Web de dados usam direta ou indiretamente pelo menos um desses itens.

Segundo [Heath and Bizer 2011] o consumo de *Linked Data* é realizado basicamente através de dois tipos de aplicações: **aplicações genéricas** que fazem uso de *Linked Data* de qualquer domínio e **aplicações de domínio específico** que são especificamente desenvolvidas para lidar com *Linked Data* relacionado a um determinado domínio.

3.4.1. Aplicações genéricas para consumo de *Linked Data*

Aplicações genéricas para consumo de *Linked Data* permitem o consumo de dados relacionados a múltiplos domínios distribuídos pelo amplo espaço de dados global. Ao percorrer os *RDF links* é possível explorar e descobrir novas informações na web de dados. A seguir serão abordados alguns tipos de aplicações genéricas normalmente usadas para acessar *Linked Data*.

Navegadores RDF

Navegadores RDF são aplicações executadas a partir dos navegadores Web convencionais que dereferenciam URIs e exibem uma visualização desse resultado, possibilitando a navegação aos dados de fontes relacionadas a partir dos *RDF links*.

*LOD Browser Switch*³⁰ é uma aplicação web que obtém detalhes a respeito de uma URI especificada pelo usuário a partir da seleção de um dos vários navegadores *Linked Data* disponibilizados pela aplicação. Assim é possível comparar a visualização de uma URI através de vários navegadores *Linked Data*.

²⁷<http://validator.linkeddata.org/vapour>

²⁸<http://vapour.sourceforge.net/>

²⁹<http://inspector.sindice.com/>

³⁰<http://browse.semanticweb.org/>

*Explorator*³¹ [Araújo and Schwabe 2009, Araújo et al. 2009] é uma ferramenta desenvolvida pelo grupo TecWeb da PUC-Rio para exploração de dados RDF através de manipulação direta. Sua interface web possibilita a obtenção de conhecimentos e respostas a questões específicas sobre um domínio através de mecanismos de visualização, busca e exploração. A ferramenta é recomendada para a formulação de consultas complexas sobre um domínio desconhecido mesmo por usuários com pouco conhecimento sobre o modelo RDF.

*Disco Hiperdata Browser*³² é uma aplicação Web usada como navegador simples para visualizar informações sobre um recurso em página HTML. Para iniciar a navegação, o usuário digita a URI do recurso em uma caixa de texto e pressiona o botão “Go”. A partir daí, *Disco* recupera as informações sobre o recurso e as exibe em uma tabela contendo propriedades, valores e fontes. As abreviações G1, G2, ..., Gn referem-se a uma lista de todas as fontes que são mostradas em outra tabela. Os *links* exibidos permitem a navegação entre os recursos, de modo que ao selecionar um novo recurso, o navegador dinamicamente recupera informações sobre ele através do dereferenciamento de sua URI. À medida que a navegação é feita, *Disco* armazena os grafos RDF recuperados em um *cache* de sessão. Ao clicar sobre o link “Display all RDF graphs”, uma nova janela é aberta contendo a lista dos grafos RDF recuperados e das URIs que não foram dereferenciadas com sucesso. *Disco* pode ser usado de forma online ou baixado para execução local. A figura 3.10 exibe informações obtidas a partir do dereferenciamento da URI http://dblp.l3s.de/d2r/resource/authors/Marco_A._Casanova pelo *Disco*.

*Marbles*³³ é uma aplicação web que funciona como navegador *Linked Data* para clientes XHTML. Pontos coloridos são usados para correlacionar a origem dos dados visualizados com uma lista de fontes de dados. Além de dereferenciar a URI, *Marbles* consulta os mecanismos de busca *Sindice* e *Falcons* por fontes que contenham informações sobre o recurso. Ele também obtém resultados de críticas relacionadas ao recurso a partir do serviço *Revyu*. Além disso, os predicados *owl:sameAs* e *rdfs:seeAlso* são seguidos para obtenção de informações adicionais sobre o recurso.

O navegador *Tabulator*³⁴ [Berners-Lee et al. 2006, Berners-Lee et al. 2007] possui os modos de exploração e análise dos dados. A utilização do modo de exploração inicia a partir da submissão de uma URI. Depois disso, *Tabulator* obtém informações sobre o recurso e as exibe como um grafo RDF em uma visão de árvore. A expansão de um nó permite a obtenção de mais informações sobre ele. Para passar ao modo de análise, o usuário pode selecionar predicados para definir um padrão e requisitar que o *Tabulator* encontre todos os exemplos daquele padrão. Para realizar a consulta, ele segue os links para encontrar o padrão no grafo RDF. Os resultados podem ser exibidos através das visões de tabela, mapa, calendário ou linha de tempo. Pode-se iniciar uma nova exploração pela seleção de um detalhe de uma das visões. *Tabulator* pode ser usado como um complemento do navegador Firefox ou como uma aplicação web que atualmente é compatível apenas com o Firefox. O principal objetivo do complemento do Firefox é explorar como dados vinculados poderiam ser visualizados em uma futura geração de navegadores Web.

³¹<http://www.tecweb.inf.puc-rio.br/explorator>

³²http://www4.wiwiss.fu-berlin.de/rdf_browser/

³³<http://www5.wiwiss.fu-berlin.de/marbles/>

³⁴<http://dig.csail.mit.edu/2005/ajar/ajaw/tab>

Marco A. Casanova

URI:

Go!

Property	Value	Sources
type	Agent ↗	G1
label	Marco A. Casanova	G1
seeAlso	http://dblp.l3s.de/Authors/Marco+A.+Casanova ↗	G1
seeAlso	http://www.bibsonomy.org/uri/author/Marco+A.+Casanova ↗	G1
retrievalTimestamp	1313004503171	G2
sourceURL	Marco A. Casanova ↗	G2
name	Marco A. Casanova	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/books/sp/Casanova81 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/adbt/CasanovaF82 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/caise/BreitmanFC05 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/cikm/BreitmanBCF07 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/dexa/HemerlyFC93 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/dexaw/LemosSC03 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/ecal/GuerreiroCH90 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/ecbs/CasanovaLLBFV10 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/ecweb/VidalC03 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/eds/FurtadoCT86 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/er/CasanovaCRL91 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/er/CasanovaTL90 ↗	G1
is creator of	http://dblp.l3s.de/d2r/resource/publications/conf/er/FurtadoCT87 ↗	G1

[next](#) 

Sources

Displayed information originates from the following RDF graphs:

- G1. http://dblp.l3s.de/d2r/resource/authors/Marco_A._Casanova [↗](#)
- G2. <http://localhost/provenanceInformation> [↗](#)

Session Cache

Display all RDF graphs that are currently in your session cache.

Figura 3.10. Visualização de informações sobre recurso através do navegador Disco

Assim, ao abrir um documento, o usuário também pode visualizar os itens que o documento descreve. As propriedades desses itens são exibidas em uma tabela e *links* podem ser seguidos para carregar mais dados sobre outros itens.

*LinkSailor*³⁵ e *Graphite RDF Browser*³⁶ são navegadores simples e rápidos para obter detalhes sobre uma determinada URI após dereferenciá-la. *LinkSailor* exibe uma visualização adaptada aos tipos dos dados exibidos.

Mecanismos de Busca *Linked Data*

O acesso à Web de Dados pode ocorrer a partir de mecanismos de busca específicos capazes de realizar pesquisas que levam em consideração a semântica dos dados. Esses mecanismos de busca permitem localizar recursos de diferentes fontes normalmente através de

³⁵<http://linksailor.com/>

³⁶<http://graphite.ecs.soton.ac.uk/browser/>

palavras-chave. A consulta pode ser realizada pelo usuário através de uma interface web ou através de serviços web providos pelos mecanismos de busca. Mecanismos de busca *Linked Data* percorrem a Web de dados percorrendo os *links* entre as fontes de dados e fornecendo a possibilidade de consultas sobre os dados dessas fontes. Os resultados das buscas são URIs que podem ser dereferenciadas e visualizadas através dos navegadores RDF. Atualmente há vários mecanismos de busca *Linked Data*. A seguir apresentaremos alguns deles.

*Sindice*³⁷ [Oren et al. 2008] coleta dados estruturados na Web (RDF, RDFa e microformatos) e os indexa por URIs, propriedades funcionais inversas (IFPs) e palavras-chave, oferecendo uma interface Web para que os usuários possam fazer buscas a partir dos itens indexados. *Sindice* também fornece um *SPARQL endpoint* que permite a realização de consultas sobre todos os seus dados e uma API para permitir a utilização de seus serviços por desenvolvedores de aplicações.

*Sig.ma*³⁸ [Tummarello et al. 2010] busca dados estruturados a partir de uma palavra-chave e os exibe em uma única página, integrando os dados de múltiplas fontes. A visão criada pelo *Sig.ma* baseia-se em resultados fornecidos pelo *Sindice*. O usuário pode aprovar, rejeitar ou acrescentar fontes para estabelecer uma visão dos dados relevantes. Ao selecionar uma entidade da lista de resultados, uma nova visão é apresentada ao usuário. Um *link* permanente pode ser criado para futuros acessos ou compartilhamento dessa visão. As filtrações das fontes de dados realizadas pelos usuários coletivamente ajudam a classificar melhor a relevância das fontes e aperfeiçoar a qualidade dos resultados futuros. Além da interface web do usuário, *Sig.ma* ainda fornece uma API destinada aos desenvolvedores de aplicações. A figura 3.11 ilustra o resultado de uma consulta sobre a pesquisadora Vânia Vidal envolvendo dezesseis fontes, onde quatro delas foram rejeitadas.

*VisiNav*³⁹ pode ser usado para consultar e navegar na web de dados através de diferentes visões (tabela, grafo, mapa, linha de tempo) sobre os dados especificados. Os dados no *VisiNav* consistem de objetos que possuem atributos ou links para outros objetos. Inicialmente uma lista de objetos é obtida a partir de palavras-chave digitadas pelo usuário. Essa lista pode ser filtrada a partir do arrastar e soltar de objetos sobre as restrições já realizadas.

*Watson*⁴⁰ [d'Aquin et al. 2007] e *Swoogle*⁴¹ [Ding et al. 2004] são mecanismos de busca mais voltados para a descoberta de informações sobre ontologias. Podem ser usados, por exemplo, para obter ontologias que possuem determinados conceitos e descobrir relacionamentos entre termos.

Outras aplicações genéricas

Informações adicionais sobre determinado recurso podem ser obtidas através da localização de objetos referenciados pelas propriedades `rdfs:seeAlso` e `owl:sameAs`.

³⁷<http://sindice.com/>

³⁸<http://sig.ma/>

³⁹<http://visinav.deri.org/>

⁴⁰<http://watson.kmi.open.ac.uk/WatsonWUI/>

⁴¹<http://swoogle.umbc.edu/>

Figura 3.11. Visão criada pelo Sig.ma sobre a pesquisadora Vânia Vidal

Serviços online de coreferenciamento como o *sameAs*⁴² são usados para encontrar URIs de diferentes fontes de dados que representam um mesmo conceito.

*LDSpider*⁴³ é um framework capaz de navegar pela web de dados seguindo *links* para obter dados de fontes *Linked Data* e os armazenar em uma *RDF Store* através de SPARQL Update ou como arquivo RDF.

3.4.2. Aplicações de domínio específico para consumo de *Linked Data*

Várias aplicações têm sido desenvolvidas para integrar *Linked Data* em domínios específicos. Essas aplicações são chamadas de *Linked Data Mashups*. A seguir descreveremos algumas delas.

*Revyu*⁴⁴ é uma aplicação web para crítica e classificação de qualquer item passível de avaliação. *Revyu* também disponibiliza uma API e um *SPARQL endpoint* para serem usados pelos desenvolvedores de aplicações.

*DBpedia Mobile*⁴⁵ [Becker and Bizer 2008] é uma aplicação cliente para dispositivos móveis consistindo de uma visão com um mapa e do navegador *Linked Data Marbles*. Baseado na localização geográfica de um dispositivo móvel, a aplicação exibe um mapa indicando localizações próximas a partir de dados extraídos das fontes *DBpedia*, *Revyu* e *Flickr*. O acesso ao *Flickr* é realizado através de um *Wrapper*. O usuário pode explorar informações sobre essas localizações e navegar em conjuntos de dados interligados. Também é possível a publicação de informações como *Linked Data*, de modo que

⁴²<http://sameas.org>

⁴³<http://code.google.com/p/ldspider/>

⁴⁴<http://revyu.com/>

⁴⁵<http://beckr.org/DBpediaMobile/>

possam ser usadas por outras aplicações.

*Talis Aspire*⁴⁶ é uma aplicação web voltada para que alunos e professores possam encontrar os principais recursos educacionais em universidades do Reino Unido. O serviço é gratuito e provê ferramentas para criar e editar listas de leitura, além da produção e publicação de materiais educativos. Quando o usuário publica conteúdo, a aplicação cria triplas RDF em uma *RDF store*. Itens publicados são interligados de forma transparente a itens correspondentes de outras instituições.

*BBC Programmes*⁴⁷ e *BBC Music*⁴⁸ são projetos desenvolvidos pela *BBC Audio and Music Interactive*. A aplicação web *BBC Programmes* disponibiliza informações detalhadas sobre tipos, séries e episódios de todos os programas de TV e rádio transmitidos pela BBC. *BBC Music* fornece informações sobre artistas, vinculando-os aos programas da BBC. Assim é possível escolher um artista e obter todos os episódios de programas relacionados a ele. As aplicações mencionadas usam *Linked Data* como tecnologia de integração de dados, inclusive fazendo uso de vocabulários amplamente conhecidos como *DBpedia* e *MusicBrainz*.

Semantic Web Pipes (SWP) [Le-Phuoc et al. 2009] apresenta um estilo de arquitetura flexível para o desenvolvimento de *mashups* de dados usando as tecnologias da Web Semântica. *Pipes* são planos de consultas criados visualmente pelo desenvolvedor do *mashup* através da conexão de operações sobre os dados usando o modelo RDF. Depois de criado, um *Pipe* é salvo e disponibilizado para uso através de uma simples requisição HTTP a uma URI específica através de Serviços Web REST.

3.4.3. APIs para manipulação de *Linked Data*

A seguir descreveremos algumas APIs para manipulação de dados na web semântica que são usadas no desenvolvimento de aplicações de domínio genérico ou específico para consumo de *Linked Data*.

*Sesame*⁴⁹ e *Jena*⁵⁰ são *frameworks* de web semântica implementados em Java que fornecem APIs para manipulação de grafos RDF.

Sesame permite armazenamento, consulta e manipulação de dados RDF. Além disso, o *framework* é extensível e configurável em relação a formas de armazenamento (memória e *RDF store*), mecanismos de inferência, formatos de arquivo RDF e linguagens de consulta (SPARQL e SeRQL).

Jena foi desenvolvido no *HP Labs* entre 2000 e 2009. Atualmente faz parte do projeto *Apache* e suas principais características são: suporte a RDF, RDFa, RDFS, OWL e SPARQL; armazenamento de triplas RDF em memória, banco de dados relacional (*Jena SDB*) ou *RDF store* (*Jena TDB*); processamento de consultas SPARQL (*Jena ARQ*); disponibilização de *SPARQL endpoint* (*Joseki* ou *Fuseki*); disponibilização de mecanismos de inferência embutidos e interfaces para mecanismos de inferência externos.

⁴⁶<http://www.talisaspire.com/>

⁴⁷<http://www.bbc.co.uk/programmes>

⁴⁸<http://www.bbc.co.uk/music>

⁴⁹<http://www.openrdf.org/>

⁵⁰<http://incubator.apache.org/jena/>

Named Graphs API for Jena (NG4J)⁵¹ é uma extensão ao framework Jena para análise, manipulação e serialização de conjuntos de grafos nomeados representando os grafos como modelos ou grafos do Jena. NG4J permite o armazenamento de grafos em memória ou em banco de dados. Consultas SPARQL podem ser realizadas sobre os grafos nomeados.

O *Semantic Web Client Library* (SWCilib)⁵² [Hartig et al. 2009] faz parte do NG4J e é capaz de representar a web de dados como um único grafo RDF. Ele recupera informações dereferenciando URIs, seguindo *links rdfs:seeAlso* e consultando o mecanismo de busca *Sindice*. O SWCilib considera todos os dados como um único conjunto global de grafos nomeados, sendo usado na implementação de vários navegadores *Linked Data*. Os grafos recuperados são mantidos em um *cache* local para melhorar o desempenho de buscas futuras.

ARQ2⁵³ é uma biblioteca escrita em PHP que contempla armazenamento de Triplas RDF, *SPARQL endpoint* e interface *Linked Data* em uma única ferramenta. As triplas RDF são armazenadas em um banco de dados MySQL. A infra-estrutura necessária para o funcionamento do ARQ2 é muito simples por requerer apenas um servidor Web com suporte a PHP e um banco de dados MySQL, sendo facilmente encontrada em qualquer serviço de hospedagem Web.

3.4.4. Abordagens para execução de consultas sobre múltiplas fontes de dados

Aplicações podem acessar *Linked Data* na web através de consultas a um *SPARQL endpoint* de um determinado conjunto de dados. Embora esse acesso possa prover dados valiosos para a aplicação, essa abordagem ignora o grande potencial da web de dados, pois não explora as possibilidades deste imenso espaço de dados que integra um grande número de conjuntos de dados interligados. Essas possibilidades podem ser alcançadas pela execução de consultas complexas e estruturadas sobre múltiplos conjuntos de dados. [Hartig and Langegger 2010] discutem diferentes abordagens para realizar essas consultas sobre a web de dados, classificando-as basicamente em dois tipos: tradicionais e inovadoras.

Abordagens Tradicionais

Data warehousing e federação de consultas são abordagens amplamente discutidas na literatura de banco de dados para realização de consultas sobre dados distribuídos em fontes autônomas. Consultas sobre a web de dados podem utilizar essas abordagens tradicionais que requerem o conhecimento prévio das fontes de dados relevantes e, portanto, limitam as fontes de dados que serão levadas em conta para obter as respostas de uma consulta. A seguir descreveremos a aplicação dessas abordagens sobre a web de dados.

Data warehousing usa uma base de dados centralizada que coleta e armazena os dados das fontes. No contexto de *Linked Data*, podem-se materializar dados das fontes relevantes em uma base centralizada para a execução de consultas sobre ela. Tal estratégia também pode ser usada em mecanismos de busca sobre a web de dados. Além disso, ela possui o melhor desempenho dentre as abordagens que serão aqui discutidas já que

⁵¹<http://www4.wiwiw.fu-berlin.de/bizer/ng4j/>

⁵²<http://www4.wiwiw.fu-berlin.de/bizer/ng4j/semwebclient/>

⁵³<http://arc.semsol.org/>

os dados podem ser acessados diretamente na base centralizada, sem a necessidade de comunicações adicionais através da rede. No entanto, em fontes de dados cujo volume de dados é muito grande, a materialização dos dados tende a requerer bastante tempo e espaço de armazenamento. Outro problema é que atualizações sobre as fontes não são imediatamente refletidas sobre o repositório central, podendo ocasionar consultas com resultados desatualizados em relação aos dados originais. Outra questão a ser considerada é que as consultas somente são realizadas sobre os dados materializados e não sobre toda a web de dados.

Federação de consultas baseia-se na distribuição do processamento de consultas para múltiplas fontes de dados autônomas. O objetivo é dar ao usuário acesso aos dados por meio de algum vocabulário padrão especificado em uma ontologia de domínio. Consultas podem ser formuladas baseadas na ontologia de domínio e um mediador transparentemente decompõe a consulta em subconsultas, direciona as subconsultas a múltiplos serviços de consulta distribuídos, e, finalmente, integra os resultados das subconsultas. Em mais detalhe, o processamento de uma consulta requer as seguintes tarefas: particionamento, adaptação, mapeamento, otimização e execução. A tarefa de adaptação consiste na modificação e extensão da consulta, por exemplo, através da inclusão de termos similares ou mais abrangentes, a partir de relacionamentos com outros vocabulários, expandindo assim o escopo do espaço de busca de forma a obter melhores resultados. A tarefa de mapeamento consiste na seleção conjuntos de *Linked Data* que têm potencial para retornar resultados para as expressões contidas na consulta. A tarefa de otimização avalia o custo de diferentes estratégias para processar a consulta, preparando um plano de execução para a consulta. Finalmente, a tarefa de execução implementa uma via de comunicação com os conjuntos de *Linked Data* e processa o plano de execução preparado pela tarefa de otimização, possivelmente adaptando-o dinamicamente. Uma vantagem da federação de consultas é que ela não requer tempo ou espaço adicional para materialização de dados. Por outro lado, a execução de consultas é mais lenta devido às transmissões de rede necessárias para realização das subconsultas sobre as fontes de dados. Além disso, as consultas não podem ser realizadas sobre toda a web de dados, mas somente sobre as fontes de dados registradas no mediador. *DARQ* [Quilitz and Leser 2008] é um mediador baseado no processador de consultas *Jena ARQ* capaz de realizar consultas distribuídas sobre a web dados. *SemWIQ* [Langegger 2010] é outro mediador que estende o *Jena ARQ* a fim de consultar a web de dados fazendo uso de estatísticas [Langegger and Woss 2009] para otimizar as consultas. [Vidal et al. 2011] apresentam um *framework* baseado em mediador de três níveis para integração de dados sobre *Linked Data*. Desafios relacionados à eficiência de consultas federadas e uma abordagem para otimização dessas consultas baseada em programação dinâmica foram tratados por [Görlitz and Staab 2011].

Abordagens Inovadoras

As abordagens inovadoras surgiram para eliminar a restrição imposta pelas abordagens tradicionais de limitarem as consultas sobre as fontes previamente conhecidas. Assim, elas permitem a descoberta das fontes durante a execução das consultas, podendo atuar sobre toda a web de dados. [Hartig and Langegger 2010] caracterizam duas abordagens inovadoras: descoberta ativa baseada em federação de consultas e consultas exploratórias (também conhecidas como *link traversal*).

Descoberta ativa baseada em federação de consultas é uma estratégia base-

ada na combinação de processamento de consultas federado com uma descoberta ativa de fontes de dados relevantes pode ser usada para possibilitar o uso de fontes de dados desconhecidas. Essa estratégia parece não ter sido implementada até o momento da publicação do presente capítulo, mas é uma estratégia que vale a pena ser objeto de investigações futuras, desde que pode combinar as vantagens da federação de consultas com a possibilidade de obter dados de fontes ainda desconhecidas pelo mediador.

Consultas exploratórias (*link traversal*). No enfoque exploratório, proposto por [Hartig et al. 2009] dados são descobertos e recuperados em tempo de execução da consulta. Este enfoque é baseado na busca de URIs, onde uma consulta SPARQL é executada através de um processo iterativo onde URIs são dereferenciadas de modo a recuperar suas descrições em RDF na Web e os resultados da consulta construídos a partir dos dados recuperados. Desse modo, consultas exploratórias seguem *RDF links* para obter mais informações sobre os dados já existentes. Através do uso de dados recuperados a partir das URIs usadas em uma consulta como ponto de partida, o processador de consultas avalia partes da consulta. Soluções intermediárias resultantes dessa avaliação parcial geralmente contêm URIs adicionais que possuem ligações para outros dados que por sua vez, podem prover novas soluções intermediárias para a consulta. Para determinar o resultado completo da consulta, o processador de consultas avalia as partes da consulta e dereferencia URIs. O conjunto de dados usado na consulta é continuamente ampliado com dados potencialmente relevantes da web, cuja descoberta é realizada a partir das URIs de soluções intermediárias que podem estar em espaços de nomes distintos. Este enfoque possui duas limitações: recupera apenas URIs e exige que a consulta seja executada a partir de uma URI somente, que faz o papel de padrão para a consulta. Finalmente, o fato do enfoque ser centralizado limita a otimização do processamento de consultas [Reddy and Kumar 2010]. SQUIN⁵⁴ [Hartig et al. 2009] é uma interface de consulta sobre *Linked Data* que implementa a abordagem de consultas exploratórias.

3.5. Limitações e Desafios

Esta seção aborda limitações das tecnologias atuais e apresenta desafios que ainda precisam ser superados para aperfeiçoar o consumo de *Linked Data*. As tecnologias atuais revelam deficiências como interfaces com o usuário ainda precárias; desempenho insatisfatório nas consultas sobre múltiplas fontes de dados; instabilidade no acesso a essas fontes; acesso a *links* quebrados e descoberta de fontes de dados relevantes. Além disso, faltam estratégias bem definidas para garantir a privacidade dos dados e tratar restrições sobre eles.

Desafios relacionados com a publicação de *Linked Data*

[Hartig and Langegger 2010] afirmam a necessidade de tornar mais transparente a integração de dados entre múltiplas fontes. Isso requer mapeamentos entre termos de diferentes vocabulários usados por fontes de dados com conteúdos similares. Além disso, pode ser necessário aplicar técnicas de fusão de dados para obter uma representação consistente de dados descritos diferentemente em fontes distintas, bem como, ajudar a resolver questões relacionadas a conflitos e qualidade dos dados. Muito ainda precisa ser feito também em relação à inferência e descoberta de conhecimento em dados provenientes de

⁵⁴<http://squin.sourceforge.net/>

múltiplas fontes.

Permitir o mapeamento dos diversos vocabulários existentes, para que seja possível identificar e escolher dados de fontes diferentes sobre uma mesma entidade também é uma questão que requer maior aprofundamento.

Permitir a criação, edição e manutenção de *Linked Data* por vários usuários é um desafio. Outro desafio está relacionado à manutenção desses dados para evitar problemas de acesso a informações que não estejam mais disponíveis. A Web de Dados é dinâmica e deve permitir que aplicações possam fazer atualizações e utilizar técnicas avançadas para a detecção de inconsistências. A web de dados é alimentada com dados provenientes dos mais diversos domínios, causando problemas quanto à confiabilidade e qualidade daquilo que é disponibilizado.

As possibilidades criadas por esses dados integrados podem infringir os direitos de privacidade dos usuários. Proteger os direitos dos indivíduos se torna difícil, pois os dados estão em fontes descentralizadas e sob diversas jurisdições legais. Prover ferramentas para explicitar os direitos de cópia e reprodução sobre os dados é uma das lacunas no contexto de *Linked Data*.

Desafios relacionados com o consumo de *Linked Data*

Já existem várias aplicações funcionais e em desenvolvimento que permitem consultas complexas na Web de Dados, porém, ainda existem muitas oportunidades de pesquisa relacionadas à forma que os usuários poderão navegar por esses dados para tornar essa interação mais intuitiva, simples e objetiva.

Há algumas formas de consulta sobre múltiplas fontes *Linked Data*. Pode-se usar materialização dos dados em uma base centralizada, consultas federadas ou consulta exploratória (*link traversal*). No entanto, ainda é necessário aperfeiçoar ou mesmo integrar esses tipos de acessos para tirar proveito das vantagens de cada um.

Determinar as informações mais relevantes, assim como detectar sua validade para melhorar a qualidade da informação, também são desafios que precisam ser superados através de algum *feedback* do usuário ou mesmo de forma automatizada.

Encontrar *SPARQL endpoints* relevantes normalmente é uma tarefa complexa. Para simplificá-la é possível obter a listagem de vários *endpoints* a partir do endereço <http://esw.w3.org/topic/SparqlEndpoints>. Além disso, a versão 1.1 do protocolo SPARQL prevê a existência do mecanismo **Descrição de Serviço** [Williams 2006] para a descoberta de informações sobre o *SPARQL endpoint*. **VoiD** (*Vocabulary of Interlinked Data-sets*) [Alexander et al. 2011] é um vocabulário usado para definição de metadados sobre fontes de dados RDF. A partir dele fica muito mais fácil identificar fontes de dados relevantes. No entanto, muito ainda precisa ser realizado para reduzir ainda mais a complexidade da descoberta dessas fontes.

3.6. Conclusão

Utilizando os mecanismos de acesso padronizados disponibilizados como *Linked Data*, é possível ter acesso a fontes de dados ilimitadas em busca de um melhor aproveitamento do potencial da web e revolucionando a forma como os dados são publicados e consu-

midos. O volume de dados disponibilizados seguindo os princípios de *Linked Data* é enorme, cresce muito rapidamente e cobre os mais variados domínios. Várias aplicações apresentadas ao longo deste capítulo utilizam estes dados para publicação e consumo de *Linked Data*. No entanto, vários desafios ainda precisam ser superados para que seja possível aproveitar todo o potencial da Web de Dados, restando ainda muito mais a ser feito. Assim, interfaces mais interativas e de fácil uso para que o usuário possa consultar e navegar pela web de dados faz falta, assim como outras questões que precisam ser mais bem desenvolvidas, como desempenho de consultas e qualidade dos dados retornados.

Referências

- [Adida and Birbeck 2008] Adida, B. and Birbeck, M. (2008). RDFa Primer — Bridging the Human and Data Webs. <http://www.w3.org/TR/xhtml-rdfa-primer/>.
- [Alexander et al. 2011] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. (2011). Describing Linked Datasets with the VoID Vocabulary. <http://www.w3.org/2001/sw/interest/void/>.
- [Araújo and Schwabe 2009] Araújo, S. F. C. and Schwabe, D. (2009). Explorator: a Tool for Exploring RDF Data Through Direct Manipulation. In *LDOW 2009: Linked Data on the Web*.
- [Araújo et al. 2009] Araújo, S. F. C., Schwabe, D., and Barbosa, S. D. J. (2009). Experimenting with Explorator: a Direct Manipulation Generic RDF Browser and Querying Tool. In *Workshop on Visual Interfaces to the Social and the Semantic Web (VISSW2009)*.
- [Auer et al. 2009] Auer, S., Dietzold, S., Lehmann, J., Hellmann, S., and Aumueller, D. (2009). Triplify: Light-weight linked data publication from relational databases. In Quemada, J., León, G., Maarek, Y. S., and Nejdl, W., editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 621–630. ACM.
- [Becker and Bizer 2008] Becker, C. and Bizer, C. (2008). DBpedia Mobile: A Location-Enabled Linked Data Browser. In *Linked Data on the Web (LDOW2008)*.
- [Berners-Lee 2006] Berners-Lee, T. (2006). Linked Data - Design Issues. <http://www.w3.org/DesignIssues/LinkedData.html>.
- [Berners-Lee et al. 2006] Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., and Sheets, D. (2006). Tabulator: Exploring and Analyzing Linked Data on the Semantic Web. In *In Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06)*, page 06.
- [Berners-Lee et al. 2005] Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform resource identifier (URI): Generic syntax. Internet Engineering Task Force RFC 3986, Internet Society (ISOC). Published online in January 2005 at <http://tools.ietf.org/html/rfc3986>.
- [Berners-Lee et al. 2007] Berners-Lee, T., Hollenbach, J., Lu, K., Presbrey, J., D’ommeaux, P. E., and Schraefel, M. (2007). Tabulator redux: Writing into the semantic web. Technical report, School of Electronics and Computer Science, University of Southampton, Southampton, UK.
- [Berrueta et al. 2008] Berrueta, D., Fernández, S., and Frade, I. (2008). Cooking HTTP content negotiation with Vapour. In *Proceedings of the 4th Workshop on Scripting for the Semantic Web 2008 (SFSW2008)*.

- [Berrueta and Phipps 2008] Berrueta, D. and Phipps, J. (2008). Best Practice Recipes for Publishing RDF Vocabularies. <http://www.w3.org/TR/swbp-vocab-pub/>.
- [Bizer and Cyganiak 2006] Bizer, C. and Cyganiak, R. (2006). D2R Server – Publishing Relational Databases on the Semantic Web. In *5th International Semantic Web Conference*.
- [Bizer et al. 2007a] Bizer, C., Cyganiak, R., and Gaus, T. (2007a). The RDF Book Mashup: from Web APIs to a web of data. In *The 3rd Workshop on Scripting for the Semantic Web (SFSW 2007), Innsbruck, Austria*.
- [Bizer et al. 2007b] Bizer, C., Cyganiak, R., and Heath, T. (2007b). How to Publish Linked Data on the Web. <http://www4.wiwiw.fu-berlin.de/bizer/pub/LinkedDataTutorial/>.
- [Bizer et al. 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked Data - The Story So Far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22.
- [Bizer et al. 2011] Bizer, C., Jentzsch, A., and Cyganiak, R. (2011). State of the LOD Cloud. <http://www4.wiwiw.fu-berlin.de/lodcloud/state/>.
- [Bizer and Seaborne 2004] Bizer, C. and Seaborne, A. (2004). D2RQ - Treating Non-RDF Databases as Virtual RDF Graphs. In *ISWC2004 (posters)*.
- [Clark et al. 2008] Clark, K. G., Feigenbaum, L., and Torres, E. (2008). SPARQL Protocol for RDF. <http://www.w3.org/TR/rdf-sparql-protocol/>.
- [Cyganiak 2007] Cyganiak, R. (2007). Debugging Semantic Web sites with cURL. <http://richard.cyganiak.de/blog/2007/02/debugging-semantic-web-sites-with-curl/>.
- [d’Aquin et al. 2007] d’Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., and Motta, E. (2007). Characterizing knowledge on the semantic web with watson. In *Evaluation of Ontologies and Ontology-Based Tools: 5th International EON Workshop*.
- [Das et al. 2011] Das, S., Sundara, S., and Cyganiak, R. (2011). R2RML: RDB to RDF Mapping Language. <http://www.w3.org/TR/2011/WD-r2rml-20110324/>.
- [Ding et al. 2004] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004). Swoogle: a search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management, CIKM ’04*, pages 652–659, New York, NY, USA. ACM.
- [Erling and Mikhailov 2006] Erling, O. and Mikhailov, I. (2006). Mapping Relational Data to RDF in Virtuoso. <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSSQLRDF>.
- [Gearon et al. 2011] Gearon, P., Passant, A., and Polleres, A. (2011). SPARQL 1.1 Update. <http://www.w3.org/TR/sparql11-update/>.
- [Görlitz and Staab 2011] Görlitz, O. and Staab, S. (2011). Federated Data Management and Query Optimization for Linked Open Data. In Vakali, A. and Jain, L., editors, *New Directions in Web Data Management I*, volume 331 of *Studies in Computational Intelligence*, pages 109–137. Springer Berlin / Heidelberg.

- [Hartig et al. 2009] Hartig, O., Bizer, C., and Freytag, J.-C. (2009). Executing SPARQL Queries over the Web of Linked Data. In Bernstein, A., Karger, D., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., and Thirunarayan, K., editors, *The Semantic Web - ISWC 2009*, volume 5823 of *Lecture Notes in Computer Science*, pages 293–309. Springer Berlin / Heidelberg.
- [Hartig and Langegger 2010] Hartig, O. and Langegger, A. (2010). A Database Perspective on Consuming Linked Data on the Web. *Datenbank-Spektrum*, 14(2):1–10.
- [Heath and Bizer 2011] Heath, T. and Bizer, C. (2011). *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition.
- [Langegger 2010] Langegger, A. (2010). *A Flexible Architecture for Virtual Information Integration based on Semantic Web Concepts*. PhD thesis, J. Kepler University Linz.
- [Langegger and Woss 2009] Langegger, A. and Woss, W. (2009). Rdfstats - an extensible rdf statistics generator and library. In *Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application, DEXA '09*, pages 79–83, Washington, DC, USA. IEEE Computer Society.
- [Le-Phuoc et al. 2009] Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., and Morbidoni, C. (2009). Rapid Prototyping of Semantic Mash-ups through Semantic Web Pipes. In *Proceedings of the 18th international conference on World wide web - WWW '09*, pages 581–590, New York, New York, USA. ACM Press.
- [Manola and Miller 2004] Manola, F. and Miller, E. (2004). RDF Primer. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>.
- [Oren et al. 2008] Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., and Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. *Int. J. Metadata Semant. Ontologies*, 3:37–52.
- [Piccinini et al. 2010] Piccinini, H., Lemos, M., Casanova, M. A., and Furtado, A. L. (2010). Wray: a strategy to publish deep web geographic data. In *Proceedings of the 2010 international conference on Advances in conceptual modeling: applications and challenges, ER'10*, pages 2–11, Berlin, Heidelberg. Springer-Verlag.
- [Prud'hommeaux and Seaborne 2008] Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query/>.
- [Quilitz and Leser 2008] Quilitz, B. and Leser, U. (2008). Querying Distributed RDF Data Sources with SPARQL. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications, ESWC'08*, pages 524–538, Berlin, Heidelberg. Springer-Verlag.
- [Reddy and Kumar 2010] Reddy, K. B. R. and Kumar, P. S. (2010). Optimizing SPARQL queries over the Web of Linked Data. In *Proceedings of the International Workshop on Semantic Data Management (SemData 2010), Singapore*.
- [Salas et al. 2010] Salas, P. E., Breitman, K. K., Viterbo F., J., and Casanova, M. A. (2010). Interoperability by design using the stdtrip tool: an a priori approach. In *Proceedings of the 6th International Conference on Semantic Systems, I-SEMANTICS '10*, pages 43:1–43:3, New York, NY, USA. ACM.

- [Sauermann and Cyganiak 2008] Sauermann, L. and Cyganiak, R. (2008). Cool URIs for the Semantic Web. <http://www.w3.org/TR/cooluris/>.
- [Tummarello et al. 2010] Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., and Decker, S. (2010). Sig.ma: Live views on the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(4):355 – 364. Semantic Web Challenge 2009; User Interaction in Semantic Web research.
- [Vidal et al. 2011] Vidal, V. M. P., de Macêdo, J. A. F., Pinheiro, J. C., Casanova, M. A., and Porto, F. (2011). Query Processing in a Mediator Based Framework for Linked Data Integration. *IJBDCN*, 7(2):29–47.
- [Williams 2006] Williams, G. T. (2006). SPARQL 1.1 Service Description. <http://www.w3.org/TR/sparql11-service-description/>.

Capítulo

4

O que sua personalidade revela?

Fidelizando clientes web através de Sistemas de Recomendação e traços de personalidade

Maria Augusta S. N. Nunes¹, Silvio César Cazella²

¹Universidade Federal de Sergipe {gutanunes@dcomp.ufs.com}

²UFCSPA/UNISINOS{silvio.cazella@gmail.com}

Abstract

Recently, studies from (Damásio 1994), (Simon 1983), (Picard 1997), (Trappl et al 2003), (Thagard 2006) and (Nunes 2009) have demonstrated how important psychological aspects of people such as personality traits and emotions are during the human decision-making process. Those studies have demonstrated how much subtle and inherent aspects from human personality have influenced the human interpersonal interaction. Indeed it proved how much those aspects could enhance the personalization during the human-human interaction and how it could be beneficial by offering products to customers in conventional business processes in the real world. Some studies (Reeves and Nass 1996) have been conducted showing that humans respond psychologically to computers and other media as if these were also human. Considering this aspect, no matter what kind of resource the computer use, the computer will be potentially making decisions and working with and for people. Thus, some understanding of the nature of human psychological aspects by computer is extremely relevant and necessary in order to improve its understanding. By improving those aspects, the computer could also improve the level of customization and optimization of their processes of decision making in order to enhance human-computer interaction and therefore the personalization on the web.

Resumo

Estudos recentes de psicólogos, neurologistas, antropólogos e cientistas computacionais (Damásio 1994), (Simon 1983), (Picard 1997), (Trappl et al 2003), (Thagard 2006) e (Nunes 2009) têm provado o quão importante os aspectos psicológicos humanos, tais como emoção e traços de personalidade, são no processo de tomada de decisão humana. Os mesmos estudos provam que esses aspectos sutis e inerentes a personalidade humana influenciam de maneira efetiva e particular suas interações interpessoais potencializando a personalização na interação humano-humano podendo substancialmente favorecer aspectos de processos comerciais convencionais na oferta de produtos e serviços no mundo real. Alguns estudos (Reeves and Nass 1996) têm sido conduzidos indicando que os humanos respondem psicologicamente a computadores e outras mídias como se esses fossem, também, humanos. Considerando

esse aspecto, não importa que recurso o computador estará usando, entretanto, em todos os casos, o computador estará, potencialmente, tomando decisões e trabalhando com e para as pessoas. Assim, para o computador, o entendimento da natureza psicológica humana é extremamente relevante e necessária para que o mesmo possa melhorar sua compreensão do ser com quem interage, melhorando assim, o nível de personalização e otimização dos seus processos de tomada de decisão visando potencializar a interação humano-computador e conseqüentemente a personalização em ambientes web.

4.1. Introdução

Atualmente o uso da *web* como uma fonte de entretenimento, cultura, informação, de produtos e serviços é, de certa forma, indispensável às atividades diárias da grande parte das pessoas nas civilizações modernas. Atualmente a *web* é considerada pelas pessoas como uma fonte inesgotável de recursos, de todo tipo, onde tudo pode ser encontrado, executado, solucionado e, principalmente, onde tudo é possível e acessível. Nesse tipo de ambiente, onde os humanos podem virtualmente viver e usar qualquer recurso real/virtual para alcançar o desejado, a personalização das informações, produtos e serviços oferecidos aos usuários é fundamental. Não importa que tipo de recurso na *web* seja utilizado, em todos os casos o computador estará potencialmente trabalhando com, e, para as pessoas. Para que a personalização ocorra de forma adequada uma eficiente forma de realizar o processo de tomada de decisão computacional deve ser adotada.

Considerando a sobrecarga de informação disponibilizadas na *web* dificilmente a personalização de informações, produtos e serviços tem se dado de forma efetiva no Brasil. Na Europa e Estados Unidos esse problema tem sido contornado pelo uso efetivo de Sistemas de Recomendação que manipulam a grande massa de informação disponível na *web* filtrando o que realmente interessa ao usuário de *e-commerce* e *e-services*. Dessa forma, a *web* brasileira vem perdendo um grande potencial mercadológico pois o empresariado vem negligenciando esse aspecto. Há uma estranha contradição nessa questão, pois a Academia brasileira produz ciência e tecnologia suficiente para inovar as técnicas utilizadas em *e-commerce* e em *e-services* no Brasil, entretanto o conhecimento produzido é subutilizado pelo empresariado brasileiro, o que acaba acarretando o déficit de nossa tecnologia comercial se comparado a Europa e Estados Unidos.

Assim, esse capítulo propõe diminuir o déficit disponibilizando um portfólio dos trabalhos no que tange tanto o estado da arte como o da técnica dos trabalhos em andamento que direcionam a área de Sistemas de Recomendação com o uso inovador da personalidade, subárea da Computação Afetiva, principalmente na linha de *e-commerce* e *e-services*.

A seguir apresenta-se a estrutura do capítulo: na seção 1.1 é discutido aspectos introdutórios do assunto proposto pelo capítulo bem como sua agenda; na seção 1.2 é introduzida a área Computação Afetiva, incluindo uma breve descrição dos aspectos que envolvem a afetividade enfatizando a personalidade; na seção 1.3, é descrito e exemplificado como, porque e quando a Computação Afetiva, principalmente a personalidade, potencializa a tomada de decisão humana. Apresenta-se, então, as abordagens de personalidade existentes exemplificando as abordagens codificáveis em computadores. Na seqüência, discute-se como os aspectos de personalidade influenciam na identidade do usuário e como isso afeta seu perfil. Seguindo-se pela discussão dos critérios de armazenamento existentes (*Markup Languages, Ontologies, User Profile*). Finalmente, são descritas as metodologias existentes hoje para extração de personalidade por computadores; na quarta seção são descritas as formas existentes de tomada de decisão computacional enfatizando os Sistemas de Recomendação, focando em sua aplicação em *e-commerce, e-services* e TV digital. Questões relativas a técnicas, e estratégias de recomendação serão apresentadas com o resumo de seus algoritmos; a seção 1.5 apresenta a concatenação das duas seções anteriores. Nessa será exemplificado como Sistemas de Recomendação podem efetivamente personalizar ambientes com

ganhos efetivos usando a personalidade; finalmente a seção 1.6 apresentada conclusões e perspectivas futuras da área de pesquisa, seguido pela seção 1.7 onde as referências bibliográficas são apresentadas.

4.2. Computação Afetiva

Desde a década de 70, cientistas computacionais, principalmente da área de Computação Afetiva buscam modelar e implementar aspectos psicológicos humanos em ambientes computacionais.

Na Computação Afetiva estuda-se como os computadores podem reconhecer, modelar e responder às emoções humanas (dentre outros aspectos) e, dessa forma, como podem expressá-las através de uma interface/interação computacional (Picard 1997). Acredita-se que permitindo que computadores expressem/captem fisiológica e verbalmente informações psico-afetivas, em uma interação humano-computador, é possível induzir e despertar afetividade em humanos. O principal objetivo de se promover esse interfaceamento afetivo é contribuir para o aumento da coerência, consistência, predicabilidade e credibilidade das reações personalizando as respostas computacionais providas durante a interação humana via interface humano-computador.

O usuário é um agente em um ambiente computacional (*web*, por exemplo) onde a interface, ou a adaptação da mesma, é fundamental para que se crie uma personalização da interação com o usuário contextualizado-o em sua zona de conforto e necessidades emergentes. Muitas características da identidade pessoal (aspectos psicológicos e habilidades sociais) do usuário podem ser captadas por meio de símbolos disponíveis na interface dos ambientes computacionais. Porém cada símbolo pode ser interpretado diferentemente por cada sujeito/usuário, considerando como o mesmo os projeta durante suas interações com o mundo real. Da mesma forma, os símbolos projetados via interface são, também, interpretados diferentemente por cada usuário, emergindo/brotando, dessa forma, diferentes aspectos psicológicos, tais como personalidade, durante sua interação no ambiente podendo influenciar definitivamente (positiva ou negativamente) na interação do usuário em ambientes computacionais atuais, tais como *e-commerce* e *e-services*.

4.3. Tomada de decisão humana & Computação Afetiva

Como descrito anteriormente, estudos recentes de psicólogos, neurologistas, antropólogos e cientistas computacionais (Damásio 1994), (Simon 1983), (Goleman 1995), (Paiva 2000), (Picard 1997), (Trappl et al 2003), (Thagard 2006) e (Nunes 2009) têm provado o quão importante os aspectos psicológicos humanos, tais como Emoção e personalidade, são no processo de tomada de decisão humano influenciando, assim, suas interações. Assim, para o computador, o entendimento da natureza psicológica humana é extremamente relevante e necessária para que se possa melhorar seu nível de personalização e otimizar a interação também em ambientes computacionais. Considerando essa necessidade, esse capítulo se propõe a tratar a questão da personalidade humana como forma de potencializar a interação humano-computador apresentando primeiramente abordagens psicológicas passíveis de codificação em computadores.

4.3.1. Personalidade

Na Psicologia não existe um consenso para a definição de personalidade. De acordo com Schultz (1990) a origem em latim da palavra personalidade “*Persona*” refere-se a máscara usada por um ator para a encenação de uma peça teatral ao público. Schultz ainda estende sua definição descrevendo personalidade como “um conjunto permanente e exclusivo de características identificáveis nas ações/interações do indivíduo em diferentes situações”. Ainda, Burger (2000) define personalidade como “um padrão de comportamento consistente e processo intrapessoal que é originado internamente no indivíduo”.

A personalidade é mais que apenas a aparência superficial e física de um indivíduo, ela é relativamente estável e previsível, porém ela não é necessariamente rígida e imutável. A personalidade, geralmente, permanece estável por um período de 45 anos iniciando na fase adulta (Soldz and Vaillant 1998). A personalidade pode ser definida segundo diversas abordagens, uma abordagem bastante interessante é a abordagem de traços de personalidade que permite diferenciar psicologicamente pessoas usando traços mesuráveis e conceituáveis. Traços de personalidade são formados por um conjunto de características humanas factíveis de modelagem e implementação em computadores (Nunes 2009).

Os traços de personalidade foram historicamente definidos por Allport (1927). Allport criou 17.953 traços (“comuns” e “individuais”) para descrever a personalidade de um indivíduo. Logo após Allport, pesquisadores assumiram que todos os homens eram identificáveis “como algum outro homem” e, dessa forma, a maioria das diferenças individuais (representadas pelos traços individuais e Allport) eram insignificantes nas interações diárias humanas e, assim eles limitaram exponencialmente o número de definições de traços. Posteriormente, os pesquisadores reduziram mais de 99% dos traços. Acabaram restando cinco fatores que se replicaram em seus estudos empíricos, como resultado, o modelo *Big Five* (John and Srivastava 1999) foi criado. Porém, mesmo considerando que o *Big Five* representasse grande eficiência na representação da estrutura de personalidade, ele não garantia exaustivamente todas as dimensões de personalidade. Dessa forma, *facet*s também foram criadas e usadas pelos psicólogos para dotar o *Big Five* de características mais detalhadas (Goldberg *et al* 2006).

Para que exista possibilidade de se personalizar a oferta e recomendar produtos, serviços em um ambiente *web* à um usuário é necessário ter-se conhecimento sobre quem é este usuário. Antes mesmo de pensar em capturar e armazenar a informações pessoais e comportamentais dele é necessário identificar que o tipo de informação será relevante para a geração da personalização adequada. Para a correta geração da recomendação a definição do perfil do usuário e tipo de informações usada é imprescindível. Abaixo apresenta-se como a personalidade influencia nessas definições através de sua influência na identidade e, conseqüentemente no o perfil do usuário.

4.3.2. Identidade e Perfil de Usuário

4.3.2.1 Identidade do Usuário

Segundo a visão da psicologia clássica, identidade é definida pela autoconsciência/visão que cada pessoa possui de si mesma, enquanto que na Psicologia Social e Sociologia, identidade pode ser definida como a forma que cada pessoa é vista sob os olhos da sociedade.

Segundo os pesquisadores de Teoria da personalidade, o desenvolvimento da identidade recebe uma importante influência da personalidade. Boyd (Boyd 2002) descreve dois aspectos diferentes da identidade: a noção internalizada do “eu” (identidade interna) e a versão projetada da internalização do “eu” (identidade social). Nessa mesma linha, Erikson (1980) por exemplo, acredita que identidade (EGO) tem uma representação pessoal interna (identidade interna) bem como uma representação social (identidade social). Giddens (1991) concorda que sem experiências sociais o “eu” não pode internalizar evolução. Giddens ainda afirma que a identidade de um indivíduo não é estática, ela pode ser representada em constante evolução, principalmente porque o componente social é dinâmico e esta sempre sendo modificado. Mead (1934) ainda define “eu” e “mim”, onde “mim” representa o aspecto socializado da pessoa (identidade social), enquanto que o “eu” representa como a pessoa se define em relação aos outras pessoas da sociedade (identidade individual).

Note que no mundo virtual onde não há presença física e conseqüentemente não há percepção de características sutis da identidade, várias pistas que possivelmente identificariam dicas de preferências, comportamentos, habilidades sociais, entre outras, são ausentes, ao contrário do que ocorre no mundo real (Donath 1999). Donath (2000) afirma que conhecer a identidade da pessoa é vital para uma adequada personalização de uma ambiente no mundo virtual, como na *web* por exemplo. Goffman (1959) afirma ainda, que as pessoas se esforçam para se apresentar como “aceitáveis” aos olhos da sociedade (em comunidades virtuais, por exemplo).

Considerando a identidade como um canal importante onde as características objetivas e subjetivas das pessoas emergem, denomina-se de fundamental importância seu uso em Sistemas de Recomendação no intuito de fornecer pistas sobre os futuros comportamentos e necessidades dos usuários em um dado ambiente onde a personalização se faz eficaz.

T tecnicamente, em Ciência da Computação, a tecnologia usada para formalizar a identidade em um dado ambiente computacional é pelo uso de Perfil/Modelo do Usuário (identidade Interna) e Reputação do Usuário (identidade Social).

4.3.2.2 Perfil de Usuário

Donath (1999) afirma que para a formação eficiente de uma identidade Virtual é crucial que o usuário tenha definida sua identidade interna e sua identidade social. No mundo virtual a identidade interna do usuário é definida por ele próprio similar ao mundo real (algumas vezes também é descoberta através de técnicas de *Machine Learning*). Enquanto a identidade social é definida pelos outros membros do mundo virtual (elucidada na próxima seção). Tanto a identidade interna, como a identidade social são armazenadas no perfil do usuário.

Perfis de usuários são conceitos aproximados, eles refletem o interesse do usuário com relação a vários assuntos em um momento particular. Cada termo que um perfil de usuário expressa é, num certo grau, características de um usuário particular (Poo et al 2003) incluindo todas as informações diretamente solicitadas a ele e aprendidas implicitamente durante sua interação na *web* (Carreira et al 2004). Fisicamente, o perfil do usuário pode ser visto como uma base de dados onde a informação sobre o usuário, incluindo seus interesses e preferências, é armazenada e pode ser dinamicamente mantido (Rousseau et al. 2004), (Poo et al. 2003).

Na *web* encontram-se muitos tipos de perfis de usuário com diferentes graus de complexidade. Eles são desenvolvidos no contexto de *e-commerce*, *e-learning* e *e-community*, por exemplo. Kobsa (2007) cria uma modelagem genérica de usuário para ser usada como uma *shell* para a criação de categorias de informação sobre o usuário objetivando personalizar as aplicações *web*. O modelo proposto por Kobsa é um dos mais reputados. Paiva e Self (1995) também desenvolveram uma *shell* de modelo de usuário chamado *TAGUS*, criado para melhor modelar os alunos para atividades de aprendizado. No *e-commerce*, (Riedl et al. 1999), (Herlocker et al. 2004), (Konstan et al. 1997), (Schafer et al. 1999) e (Schafer et al. 2001), do GroupLens, criaram vários modelos de usuário baseado em ranqueamento de filmes, de notícias, entre outros. Esses modelos têm sido usados nos Sistemas de Recomendação criados pelo GroupLens.

Considerando ainda definições de modelo de usuário, Heckmann (2005) e Heckmann e Kruguer (2003) propõem uma ontologia¹ de um modelo de usuário geral (*GUMO*). O *GUMO* é um modelo ubíquo de modelo de usuário incluindo muitos aspectos básicos de usuário, partindo desde informação de contato, demográficos, habilidades fisiológicas e psicológicas, estado emocional,

¹Uma ontologia é uma especificação de um conceito.

estado mental e nutrição. A ontologia de Heckmann é muito rica e pode ser implementada de acordo com o interesse do projetista de uma *shell* de perfil de usuário.

Note que para gerar as recomendações e personalizar o ambiente ao usuário, os Sistemas de Recomendação necessitam da identidade interna do usuário que é definida pelo perfil de usuário, como foi apresentado. Porém necessita, muitas vezes, também, da identidade social que é definida pela reputação do usuário.

4.3.2.3 Reputação

Reputação pode ser definida como o retorno social recebido sobre a personalidade de alguém. A reputação pode ser compatível ou não com a descrição feita no perfil de usuário. Josang et al (2007) descreve reputação como “a informação normalmente dita ou crível sobre as características de uma pessoa ou coisa e seus estados”.

Resnick et al. (2000) define reputação como a coleção dos *feedbacks* recebidos sobre o comportamento efetuado pelos participantes de uma comunidade. A reputação ajuda as pessoas a escolher parceiros confiáveis no mundo virtual que são credíveis no mundo real. Geralmente nas redes de reputação, os usuários encorajam os comportamentos confiáveis discriminando a participação de pessoas desabilitadas moralmente ou desonestas.

Segundo Rein (2005) a reputação pode ser também definida como um completo sistema de informações de opiniões alheias sobre um usuário, que inclui todos os aspectos de um modelo de referência. Esse modelo de referência é baseado em nove aspectos determinantes: conhecimento, experiência, credenciais, endosso, contribuições, conexões, sinais, *feedback*, contexto e valores sociais. A visão estrutural de Rein descreve as funcionalidades e comportamentos essenciais do ser humano que são desejáveis e efetivos para possivelmente ser representado através de uma reputação explícita e fácil de ser medida no usuário.

A reputação é geralmente aplicada para gerenciar comportamento do usuário durante um processo comercial (*e-commerce*, por exemplo) envolvendo compra e venda de produtos e/ou serviços e também durante processos sociais como combinação social em comunidades virtuais e redes sociais.

Em processos comerciais, como por exemplo, no *eBay* (Resnick et al. 2000) (Resnick et al. 2006) um consumidor compra um certo produto de alguém. Depois disso, ele deixa um *feedback* sobre o produto comprado e/ou o comportamento do vendedor durante o processo de venda.

Em contraste, em situações sociais como, por exemplo, *Orkut*, *IKarma*, *Opinity*, *LinkedIn*, *Mendeley* (Jensen et al. 2002), usuários são membros de comunidades virtuais ou redes sociais. Eles são capazes de coletar gerenciar e promover reputação de usuário entre seus clientes e contatos da comunidade ou rede. Isto é, usuários (prestadores de serviço) que tem *profile* na rede de reputação, que é também uma rede social podem ser “*tagged*” e rankeados pelos seus clientes e/ou contatos. Usuários podem ser encontrados através de *tags* em e-mail ou, também, alguém pode encontrar um contato de um prestador de serviço simplesmente procurando em *tags* na própria rede de reputação.

Nunes (2009) convencionou reputação como uma extensão de um perfil de usuário. Teoricamente se usa o mesmo tipo de informação armazenada no perfil de usuário, porém o conjunto de informações é fornecido por outro alguém (amigo, cliente do usuário, entre outros). Nesse caso, a identidade é determinada pelos traços de personalidade do usuário informados (ou automaticamente medidos) por ele mesmo para perfil de usuário e informados (ou automaticamente medidos) por uma outra pessoa para reputação de usuário.

O perfil de usuário é muito importante para definir a identidade do usuário. Dessa forma o perfil pode prever necessidades e comportamentos do usuário em um ambiente computacional, enquanto a reputação permite a criação de relação de confiança entre membros de uma comunidade

em um ambiente *web*, especialmente em *e-commerce*. A identidade do usuário é muito útil para sua interação social no ambiente computacional.

A seguir, apresentam-se os esforços dos cientistas de Ciência da Computação e Computação Afetiva com o objetivo de padronizar uma forma de armazenamento e padronização da informação afetiva já definida no perfil do usuário e/ou reputação.

4.3.3. Mecanismos para armazenamento da personalidade em computadores

Considerando a grande gama despadronizada de representação afetiva, alguns pesquisadores pertencentes ao W3C Emotion Incubator Group (W3C 2010a) têm ampliado esforços para definição de uma padronização formal da afetividade através de uma linguagem de marcação. A padronização afetiva tem focado, principalmente, na representação de emoções em uma *markup language*, considerando que emoção é o campo da Computação Afetiva historicamente mais desenvolvido. Após esforços iniciais de Heckmann (2005) na criação de uma ontologia completa incluindo diversos aspectos de Personalidade, a partir de 2008, alguns esforços pontuais de pesquisadores, tais como Nunes (2008; 2009; 2010b) tem seguido na direção duma padronização específica modelada através de uma *markup language* intitulada de PersonalityML.

4.3.3.1 XML & Markup Languages

Uma linguagem de marcação é uma forma de definir e identificar a estrutura e o significado do conteúdo em um documento. A especificação XML define um padrão para adicionar marcação a documentos, genericamente, o XML nem define semântica nem um conjunto de elementos pré-definidos, como o HTML. Na realidade, a XML pode ser considerada como uma meta-linguagem para definir linguagens de marcação, fornecendo mecanismos (através de sua especificação) para que se defina elementos e relacionamentos entre eles, para semânticas preconcebidas.

O uso do padrão XML (W3C 2010d) para definição de outras linguagens de marcação é algo amplamente desenvolvido pela comunidade acadêmica (e são extensamente utilizadas comercialmente). Alguns casos fazem parte da própria tecnologia XML, por exemplo, XML Schema e XSLT. No primeiro caso, o padrão XML Schema possui elementos pré-definidos que definem como construir esquemas para instâncias de documentos XML. No caso da XSLT, os elementos definem como um processador deve ler uma instância XML e transformá-la para outro formato texto como, por exemplo, um HTML, um documento em LaTeX ou até mesmo um simples txt. Outros exemplos são: Simple Object Access Protocol - SOAP (W3C 2010b), *Attention Profiling Markup Language* - APML (Angell et al 2010), Extensible MultiModal Annotation markup language - EMMA (W3C 2010c), Emotion Markup Language - EmotionML(W3C 2010a) entre diversas outras².

4.3.3.2 EmotionML & PersonalityML

A EmotionML (*Emotion Markup Language*) versão 1.0 foi apresentada pelo W3C (*WWW Consortium*) em outubro de 2009. Concebida como uma extensão da linguagem de marcação XML, em uma primeira tentativa da comunidade científica para padronizar a representação de emoções. A EmotionML provê elementos para uma linguagem de marcação que chama a atenção ao equilibrar a fundamentação científica e a aplicação prática. A EmotionML é uma linguagem dinâmica, ainda que muito jovem e com uma representação ainda bastante genérica. Apesar de flexível, a EmotionML não leva em consideração outros aspectos que afetam significativamente a correta identificação de uma emoção, como por exemplo a Personalidade.

² Consultar http://en.wikipedia.org/wiki/List_of_XML_markup_languages para uma lista não exaustiva.

Na realidade, apesar de existência de alguns avanços significativos quanto à Personalidade, são relativamente poucos e incipientes os trabalhos que lidam computacionalmente com este aspecto psicológico, e menos ainda aqueles que lidam com seu reconhecimento automático ou mesmo representação computacional (Mairesse et al 2007), (Heckmann 2005), (Nunes 2009), (Nunes et al 2010).

Como descrito anteriormente, pesquisadores em Computação Afetiva têm implementado as emoções explicitamente, mas de forma despadronizada antes do surgimento o da EmotionML. A representação da emoção, ao invés de outros aspectos de Computação Afetiva se tornou possível devido às emoções serem mais facilmente mensuráveis e interpretáveis e poder efetivamente influenciar diretamente na ação-interação dos usuários. As emoções são instantâneas, elas têm uma vida curta, volátil e mudam constantemente, diferentemente da personalidade que é um estado muito mais estável e, normalmente, mantido durante um período de 45 anos. Apesar disso, com base nas pesquisas é possível dizer que a personalidade implica em emoções (Lisetti 2002); cada pessoa ou agente que tem emoções tem uma personalidade; e, geralmente, a personalidade não aparece explicitamente mesmo que influencie as emoções diretamente. Lisetti (2002) descreve um modelo complexo para representar aspectos psicológicos em agentes inteligentes (virtual/real) que interagem socialmente, denominado *Affective Knowledge Representation* (AKR - Representação do Conhecimento Afetivo). No AKR apresenta-se a Personalidade como o topo do modelo hierárquico dos aspectos psicológicos, denotando assim seu maior poder.

Dessa forma, considerando que a personalidade é mais abrangente e implica na emoção, Nunes et al (2010b) propõem uma extensão à representação padronizada de emoção incorporando uma nova proposta intitulada de PersonalityML, que é também baseada em XML. Essa extensão objetiva representar a complexidade afetiva descrita brevemente em (Nunes et al 2010b), onde a personalidade é o ponto chave da cadeia e, que sua representação está limitada pela atual versão da EmotionML.

Entretanto, antes de se pensar em uma linguagem de marcação para a personalidade, deve-se lembrar que nem todas as teorias de Personalidade existentes possuem uma estrutura passível de representação e, conseqüentemente, de implementação em computadores. Felizmente há algumas que possuem, sendo as mais utilizadas na literatura da Computação Afetiva, àquelas pertencentes à abordagem dos Traços, teoria esta que serviu de base para o lançamento da primeira versão da PersonalityML (Bezerra et al 2011) sendo modular suficiente para a incorporação automática de novas teorias e abordagens existentes hoje. Dessa forma, após a formalização e criação de mecanismos de armazenamento, abaixo apresenta-se como a personalidade pode ser extraída do usuário.

4.3.4. Extração de personalidade

Considerando a abordagem de traços, escolhida por ser a melhor forma para representação de personalidade em computadores, psicólogos geralmente usam questionários intitulados de inventários de personalidade. Esses inventários são diretamente aplicados por psicólogos ou encontrados na *web*. Os mesmos podem ter uma pequena ou grande quantidade de questões e o número de questões é diretamente proporcional a granularidade e precisão dos traços de personalidade extraídos do usuário. Segundo Gosling (2008), os longos e mais precisos inventários tomam um tempo bastante considerável do usuário e muitas vezes torna inviável sua aplicação, nesses casos opta-se pelo uso de testes compactos como o TIPI (estando ciente da limitação quanto a precisão das respostas) ou mesmo opta-se pelo uso de outras formas de extração de personalidade, muitas vezes em estágio embrionário de desenvolvimento. Inicialmente apresenta-se ao leitor a forma clássica de extração de personalidade seguindo-se por abordagens embrionárias e direções de pesquisa:

4.3.4.1 Extração de Personalidade através de inventário baseado em Traços

Existem diversos inventários validados, como apresentado em (Nunes 2009; 2010). Porém um Teste de personalidade bastante interessante é o NEO-IPIP (Johnson 2000), (Johnson 2005) desenvolvido em conjunto com o *International Personality Item Pool* (Goldberg *et al* 2006). Ele permite medir as cinco dimensões do Big Five incluindo mais seis facetas para cada dimensão (30 facetas no total) usando uma descrição detalhada dos traços de personalidade humana e por conseqüência propiciando uma grande precisão na representação da personalidade.

Segundo Gosling (2008) “deixamos pistas sobre nossa personalidade em tudo o que fazemos, em nossos objetos, onde vivemos”. Sendo assim, através dos padrões de navegação de cada usuário, é possível adquirir características psicológicas, através de uma abordagem implícita e transparente ao usuário. Um pensamento natural é que esta seria a melhor forma de obtenção dos traços de personalidade dos usuários, uma vez que esta abordagem exige menor esforço cognitivo se comparado aos tradicionais inventários de personalidade. Porém, Dumm *et al* (2009) negaram esta hipótese ao pesquisar três interfaces de obtenção de traços de personalidade, sendo duas delas explícitas e uma implícita. O resultado foi que a interface NEO (utilizada nos testes como NEO-PI-R e NEO-IPIP) obteve melhor *feedback*, tanto em termos de resultados apresentados quanto em termos de facilidade de uso. Dessa forma, o teste NEO-IPIP, citado acima, torna-se, então, uma opção interessante a ser utilizada como ferramenta de entrada de dados explícita para obtenção dos traços de personalidade dos usuários. Este teste possui um *feedback* positivo em mais de 99% dos casos (Johnson 2005).

Assim, a equipe da autora propôs uma nova interface ao NEO-IPIP, intitulada *Personality Inventory*, oferecendo mais usabilidade ao ambiente proposto originalmente por Johnson (2000) e Gosling (2008). As medições de personalidade extraídas no inventário seguem os padrões propostos por Johnson (2000) e Gosling (2008). Para que os resultados pudessem ser mantidos em um local seguro e persistente e serem usados em diversas aplicações de pesquisa, optou-se por uma aplicação *web*. A interface é simples como apresentado na Figura 1a. Os inventários disponíveis estão apresentados na Figura 1b.

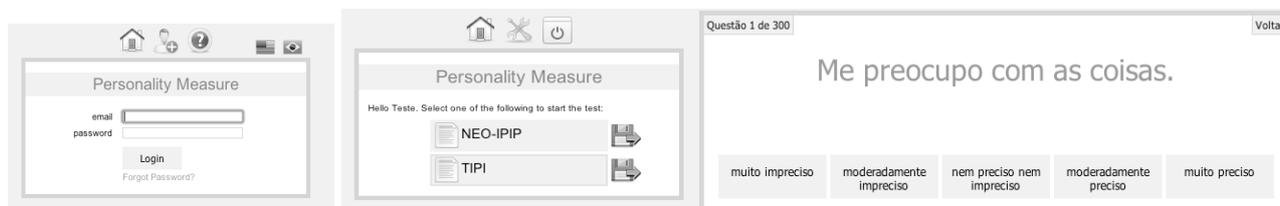


Figura (1a) Interface inicial (1b) Inventários Disponíveis (1c) Questão 1 do inventário

Como visto na figura 1a, para que a medição de personalidade seja feita, o usuário precisa criar uma conta de usuário e responder pelo menos um dos questionários disponíveis, no caso, o NEO-IPIP, como apresentado na figura 1b. Após a seleção do Inventário NEO-IPIP o usuário deve responder o questionário como apresentado na figura 1c.

Uma vez respondido o questionário do NEO-IPIP, cada questão respondida pelo usuário tem valor atribuído entre 1-5. Ao finalizar o teste, os valores atribuídos a cada uma das questões respondidas são utilizados para calcular o resultado. No cálculo, o resultado é normalizado e é atribuído um valor entre 1-100 para cada um dos itens do Big Five, bem como para as suas facetas. Note que somente então o relatório descritivo dos traços de personalidade do usuário é gerado e disponibilizado exclusivamente a ele. Na Figura 2, parte do relatório descritivo dos traços de personalidade do usuário, extraído do inventário NEO-IPIP, é apresentado.

Uma vez visualizado seu prognóstico, o usuário poderá re-visualizar o resultado através da própria aplicação ou exportá-lo como PersonalityML.

4.3.4.2 Extração de Personalidade através de outras técnicas

Note que Gosling (2008) afirma que a melhor forma de obtenção dos traços de personalidade dos usuários é através do uso de uma abordagem que não exija esforço cognitivo se comparado aos tradicionais inventários de personalidade, como o exemplo mostrado acima. Andrade et al (2011) afirma que os traços medidos através de inventários de personalidade muitas vezes são, em parte, um conjunto de dados provenientes do auto-relato da própria opinião do usuários podendo desvirtuar da sua real personalidade.

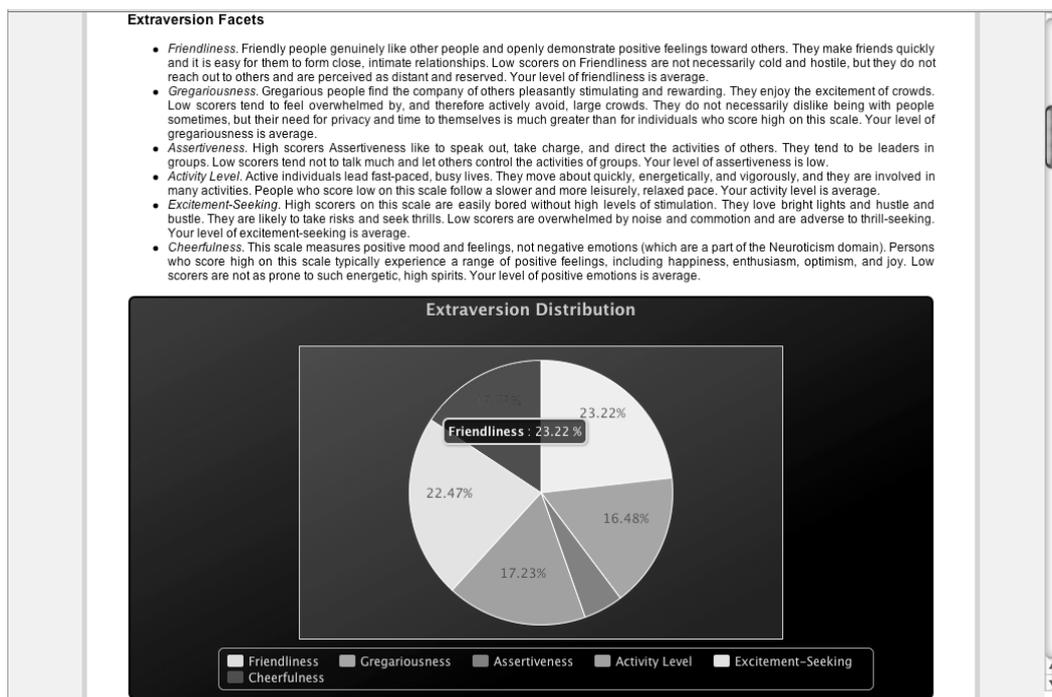


Figura 2. Relatório descritivo parcial do NEO-IPIP (Nunes 2010)

Porém, infelizmente, em computadores, ainda poucas técnicas de extração de traços de personalidade, que diferem dos tradicionais inventários, tem sido desenvolvidas e implementadas. Segundo Brinkman e Fine (2008) constantemente pesquisadores tem se aventurado em novas pesquisas que tentam obter personalidade de uma forma menos intrusiva que os tradicionais inventários. Segundo Porto et al (2011) tem-se realizado tentativas através da análise de gravações automáticas de dados da interação do usuário com o sistema, tais como a avaliação a escolha dos usuários para customização de algumas aplicações como *Windows Media Player*, *ICQPlus* e o *Winamp*, por exemplo.

Andrade e Nunes (2011) afirmam ainda, que nesse novo contexto, incluem-se as técnicas de Reconhecimento de Padrões, baseadas em cálculos probabilísticos visando reconhecer emoções ou mudanças de comportamento através da observação de um usuário utilizando o determinado sistema em uso no computador, seja pelo clique do *mouse*, por teclagem, captura de imagem do usuário pela *webcam* ou, ainda, por sensores que capturam sinais vitais dos usuários (as principais referências são os trabalhos de (Chanel 2009), (Hu and Pu 2009), (Tkalčič 2010) e (Khan et al 2008)). Rabelo e Nunes (2011) ainda afirmam que é possível identificar traços de personalidades a partir de definições de modelos e Frameworks que utilizam recursos estatísticos para classificar

determinado conjunto de dados, definido com conjunto de características, como pertencentes a um determinado traço de personalidade, associada a classes no reconhecimento de padrões. Autores como Mairesse e Walker (2005; 2008); Hussain e Calvo (2009) ; Pianesi et al (2008) convergiram as suas técnicas e modelos à identificação de personalidade a partir de extratos de diálogos de diversas modalidades, tais como: comunicação textual assíncrona (*e-mails* e dialogo em sites de relacionamento), conversa falada (áudio presencial ou virtual), expressões faciais (videoconferência), sinais fisiológicos, dentre outros. Ainda Gill e Oberlander (2003) alegam a possibilidade de detecção de traços de personalidade em textos de comunicação assíncrona, mas explicitamente as mensagens de correio eletrônico através de processos estatísticos.

Abaixo, de acordo com, Rabelo e Nunes (2011) e Porto et al (2011) alguns trabalhos que envolvem extração de traços de personalidade, são melhor detalhados:

- Em Porto et al (2011), a extração de traços de personalidade é realizada através do teclado é feita obtendo-se informações de como o usuário digita determinado texto, essa informação pode ser obtida através da latência entre teclas consecutivas (o tempo entre o apertar de uma tecla e da subsequente). Esse tempo é obtido através de um evento de teclado, o *keyPress*. O evento aciona um *timer* que é interrompido quando o evento é acionado novamente, ou seja, o *timer* conta o tempo decorrido entre uma tecla pressionada e outra, o tempo de latência. Tendo a informação de como um usuário digita, este pode ser comparado com um banco de dados que contem informações de outros usuários, informações estas que são o ritmo de digitação e a personalidade do usuário, e através de técnicas de agrupamento de dados (*clustering*), o usuário é isolado em um grupo que tem o mesmo padrão de digitação, e então os traços de sua personalidade são inferidas a partir dos usuários que fazem parte do mesmo grupo. Os testes realizados para essa abordagem tendem a confirmar que quando os usuários são agrupados levando em consideração o ritmo de digitação, alguns traços de personalidade são correlacionados. Usuários que são agrupados num mesmo cluster tendem a ter o mesmo valor para algumas facetas, e, portanto usuários que não se soubessem o valor das facetas, tendo sido agrupados em um cluster, poderiam ter seus valores inferidos pelos outros elementos do mesmo cluster.
- Gill e Oberlander (2003) investigam o impacto da iteração mediada por computador na percepção pessoal, em particular as características importantes para a socialização e colaboração, Extroversão e Neuroticismo, determinando que tais características podem ser detectadas a partir do texto de um e-mail e para a determinação das características linguísticas da personalidade destes, um conjunto de análises e técnicas foram aplicadas: Os dados LIWC(Linguistic Inquiry Word Count), base de dados e propriedades psicolinguísticas de derivados do *Medical Research Council*, além da medida de diversidade léxica conhecida com *Type Token Ratio* (TTR).
- Por sua vez, Mairesse e Walker (2005) propõem em seu trabalho uma modelagem para reconhecimento de personalidade em conversação, intitulado de *Personality Recognizer*, a partir de uso de modelos estatísticos não lineares para classificação baseada em traços de personalidade do Big Five. A abordagem pode ser resumida em cinco etapas: (1) Recolher individual dos Corpora (Textos); (2) Coletar informações de personalidade para cada participante; (3) Extrair características relevantes a partir dos textos; (4) Construir modelos estatísticos das avaliações de personalidade com base nas características; (5) Testar os modelos aprendidos nas saídas linguísticas dos indivíduos invisíveis. Os dados consistem extratos de conversas diárias de 96 participantes, utilizando um gravador ativado eletronicamente durante o período de dois dias. Para preservar a privacidade dos participantes, bits aleatórios de conversa foram registrados, e somente as afirmações dos participantes foram

transcritas, tornando-se impossível reconstruir conversas inteiras. O corpus contém 97.468 palavras e 15.269 declarações e o experimento foi dado através das etapas seguinte: (1) Seleção de Recursos: Recursos são automaticamente extraídos de cada trecho e é calculada a proporção de palavras em cada categoria a partir da ferramenta LIWC (Linguistic Inquiry Word Count), correlacionando ao Big Five. Estas características psicológicas adicionais foram computados pela média de contagens de palavras características de um banco de dados psicolinguística MRC; (2) Modelos Estatísticos: O treinamento dos modelos foi feito utilizando o algoritmo RankBoost, que classifica de acordo com cada característica do Big Five expressando a aprendizagem dos modelos como regras, que suportam a análise das diferenças nos modelos de personalidade, onde para cada extrato da conversa, a regra modifica a pontuação e assim o ranking. Os resultados obtidos são os aspectos caracterizam a produção de linguagem: (1) Tipos de expressão, conteúdo e sintaxe (LIWC); (2) As estatísticas psicolinguísticas (MRC); (3) Prosódia. Para avaliar como cada conjunto de recursos contribui para o resultado final, foram treinados os modelos com o conjunto completo de recursos e com cada conjunto, individualmente e testes T pareado mostram que os modelos de extroversão, afabilidade, consciência e abertura ao novo usando todos os recursos são melhores do que a linha de base aleatória. O Neuroticismo é o traço mais difícil de modelar, mas é melhor predito por características MRC. As características LIWC têm desempenho significativamente melhor do que a linha de base para todas as dimensões. As características prosódicas são boas predictoras de extroversão.

- Finalmente no trabalho estudado, Pianesi et al (2008) propõe um modelo para reconhecimento multimodal de traços de personalidade em interações sociais, que consiste em criar classificadores capazes de prever os traços de personalidade, considerando o comportamento de um sujeito em um minuto de uma janela temporal. Como classificador, foi utilizado Maquinas de Vetor de Suporte (SVM), que tenta encontrar um hiperplano que não só discrimina as classes, mas também maximiza a margem entre elas. Como experimento, pressupõe-se que a personalidade aparece no comportamento social e é esperado que características audiovisuais fossem apropriadas para constituir um sistema automático de exploração e classificação de traços de personalidade. Duas abordagens são descritas: Na primeira abordagem, leva-se em consideração apenas o comportamento do sujeito é suficiente. Ex.: Maneira como se move entonação da voz e etc. Na segunda leva-se em consideração além do comportamento do sujeito, também o contexto social. Foram testadas as seguintes dimensões: Características Acústicas e Características Visuais e divididas em Níveis distintos: ALL(Todas as características Acústicas + Todas as Características Visuais), SEL (Seleções de características Acústicas + Seleções de Características Visuais) e No-Feat (Todas as características Acústicas + Características Visuais ; Características Acústicas + Todas as Características Visuais, etc). A Combinação destes níveis , exemplo (ALL , No-Feat) corresponde a um vetor de características utilizadas para treinar e testar os Classificadores que serão responsáveis por definir as distribuições dos dois traços de personalidade determinados como classes: A Extroversão e O Locus de Controle. Foram comparados os resultados das classificações com um classificador trivial que atribui a classe mais frequente para cada instancia. Os resultados do experimento determinaram que, duas análises de Variância, uma para cada tipo de personalidade mostram que os principais efeitos são significativos para $p < 0,0001$. Os efeitos para Extroversão não foram significativos em $p > 0,05$, enquanto os efeitos para Locus de Controle foram significativos em $p < 0,05$.

4.4. Tomada de decisão computacional & Sistemas de recomendação

4.4.1. Sistemas de Recomendação

Segundo Resnick e Varian (1997) muitas vezes é necessário fazer escolhas sem um grande conhecimento das possibilidades que nos são apresentadas. Normalmente nos baseamos nas recomendações de amigos, através de opiniões de especialistas ou, ainda, a partir de guias. Os Sistemas de Recomendação ajudam e aumentam este processo social natural já existente entre as pessoas.

Estes sistemas se tornaram uma importante área de pesquisa com os primeiros trabalhos sobre filtragem colaborativa e filtragem baseada em conteúdo em meados da década de 90 (Adomavicius e Tuzhilin, 2005). O interesse nesta área continua grande por ela ser rica em problemas de pesquisa e pela quantidade de aplicativos desenvolvidos para ajudar os usuários a lidar com a imensa demanda de informação, gerando recomendações personalizadas, conteúdos e serviços para os mesmos.

De acordo com Cazella et al (2010), as pessoas têm acesso a uma vasta gama de informações devido à grande oferta e aos recursos da Internet, porém dispõem muito tempo na busca do que realmente é interessante ou útil para elas. A dificuldade de encontrar a informação correta é aumentada quando a informação disputa a atenção de uma pessoa com uma série de outras informações não tão relevantes.

Resnick e Varian (1997) afirmam ainda que um sistema típico agrega e direciona as avaliações de itens feitas pelos usuários e as disponibiliza como recomendações para os indivíduos considerados potenciais e interessados neste tipo de recomendação. O grande desafio está em descobrir o relacionamento de interesses entre os usuários, realizando desta forma o correto casamento entre os itens que estão sendo avaliados e os usuários que estão recebendo a referida recomendação. O sistema gerará respostas mais valiosas quando as preferências dos usuários diferirem uma das outras do que quando temos uns poucos especialistas.

4.4.2. Técnicas de recomendação

Ao contrário da recuperação de informação, onde o usuário necessita informar sua necessidade de informação e é o responsável pelo início da interação, as técnicas de filtragem buscam informações relevantes geralmente através de um perfil de interesses do usuário. A seguir serão descritas algumas técnicas de recomendação (filtragem de informação) aplicadas a Sistemas de Recomendação.

4.4.2.1. Filtragem Baseada em Conteúdo

A filtragem baseada em conteúdo (FBC) realiza uma seleção baseada no conteúdo dos itens (Adomavicius e Tuzhilin 2005). Seu objetivo é gerar de forma automatizada descrições sobre o conteúdo dos itens e compará-las com os interesses do usuário (Herlocker 2000; Ansari 2000). Através disso, pode-se verificar qual a correlação entre o que o usuário está buscando e o item apresentado (Balabanovic e Shoham 1997).

A limitação da técnica está diretamente interligada com os objetos que este sistema necessita recomendar. Portanto, é necessário que o conteúdo dos itens seja computado automaticamente ou, então, será preciso que estas informações sejam adicionadas manualmente, como por exemplo, com a associação de descritores (*tags*). Enquanto existem técnicas que funcionam bem com o processamento de documentos de texto, alguns outros domínios têm problemas com a extração automática. Exemplos são: informação multimídia, imagens, áudio e vídeo (Ansari 2000).

Um grande desafio nessa área é o tratamento de metadados. A forma de identificar o conteúdo dos atributos e conseguir inferir que, por exemplo, “gripe” tem o mesmo significado de “resfriado” em determinado contexto. Para solucionar este problema, tem sido amplamente utilizada a técnica de indexação de frequência de termos (*term frequency indexing*) (Herlocker 2000).

Neste tipo de indexação, as informações sobre o item e as necessidades do usuário são descritas por vetores com uma dimensão para cada palavra da base de dados. Cada componente do vetor representa as repetições da respectiva palavra no item. Assim, os metadados serão ordenados por frequência e quanto mais próximos estiverem ao índice, mais relevante o mesmo será. Obviamente, métodos de pré-processamento de texto são necessários para que palavras muito usadas em notícias e artigos (tais como preposições, artigos e conjunções) não sejam as palavras com maior frequência, prejudicando a predição de conteúdo.

Um problema que pode ocorrer é não conseguir distinguir dois itens caso sejam representados pelo mesmo grupo de características. Portanto, se dois artigos forem representados pelas mesmas palavras, não será possível mensurar a qualidade dos documentos (Ansari 2000).

Os usuários têm a liberdade de poder mudar de opinião e de hábitos. Sistemas de recomendação usando FBC podem tornar-se superespecializados quando o sistema já aprendeu demais sobre os hábitos daquele usuário, e não conseguirem sugerir coisas que fujam desse padrão. Um bom exemplo é o caso de um usuário que consome carne e que se tornou vegetariano. Não importa quanto tempo passe, o sistema vai continuar indicando carnes para compra. Este problema também não possibilita ao usuário receber recomendações de itens que ele já tenha gostado. Desta forma, uma pessoa que nunca tenha experimentado cozinha grega, jamais irá receber a sugestão de visitar o melhor restaurante grego da cidade (Ansari 2000).

Segundo Adomavicius e Tuzhilin (2005), uma forma de tentar resolver o problema da superespecialização é sugerir, aleatoriamente, alguns itens ao usuário. Adicionalmente, deve-se prestar atenção nos itens que são muito similares aos anteriormente vistos, pois o usuário deve receber recomendações homogêneas. Por exemplo, não é necessariamente uma boa idéia recomendar todos os filmes de Woody Allen para alguém que tenha gostado de um deles.

O problema do novo usuário também está presente nesta técnica. Acontece quando o sistema não consegue entender as preferências do mesmo, pelo fato de não ter avaliado uma quantidade mínima de itens, não gerando nenhuma recomendação. E, ainda, caso o usuário não possua muitas recomendações, a qualidade destas também pode ser baixa.

4.4.2.2. Filtragem Colaborativa

Para Cazella (2006), a essência dos sistemas colaborativos está na troca de experiências entre as pessoas que possuem interesses comuns. Para tanto, ao invés do conteúdo dos itens, a filtragem é baseada nas avaliações feitas pelos usuários daqueles itens.

A filtragem colaborativa (FC) foi proposta para suprir as necessidades da filtragem baseada em conteúdo (Herlocker 2000). Desta forma, os usuários receberão recomendações de pessoas com gostos similares e que gostaram do item no passado, enquanto que na FBC serão recomendados itens similares aos quais o usuário gostou no passado. É um processo que busca usuários similares e calcula, baseado na semelhança entre os mesmos, qual a nota aproximada que o usuário em questão daria ao produto caso o fizesse. Esta técnica também é chamada de “*k-nearest-neighbor*” ou “*user-based*” (Herlocker 2000).

Um ponto fraco da FC é o seu desempenho. Ela tem de ser rodada *online*, e caso seja executada para todos os usuários, leva-nos a um desempenho de $\Omega(N^2)$ (limite assintótico inferior). Uma heurística poderia ser aplicada no passo intermediário, descartando vizinhos que fogem de um

padrão pré-estabelecido ou que não possuam a nota que se busca prever (a fórmula de correlação necessita que todos os vizinhos utilizados possuam a nota do produto que se busca fazer a previsão). De acordo com Adomavicius e Tuzhilin (2005), pode-se, também, calcular a similaridade de todos os usuários com antecedência e recalculá-la apenas de vez em quando, pois os pares não costumam mudar drasticamente em um curto espaço de tempo.

Também está presente nesta técnica o problema do cold start ou novo usuário. Quando este não possui muitos itens avaliados, o sistema não consegue encontrar usuários similares. Isso se deve ao fato do cálculo da correlação do coeficiente de Pearson necessitar que todos os usuários vizinhos tenham itens em comum para a previsão de um item. Algumas estratégias foram propostas para superar esse problema, como a criação de agentes autônomos e a utilização de promoções ou alguma forma de incentivo para aumentar a participação do usuário. Dificuldade similar enfrenta o novo item quando há a sua inclusão no sistema e não é possível recomendá-lo até que algum usuário avalie o mesmo.

Os sistemas baseados em FC também são muito suscetíveis à falhas, segundo Sandvig et al (2007), pois sua busca é baseada em usuários singulares que possuam similaridade, não havendo nenhuma filtragem prévia para verificar se o usuário é um ruído na base de dados. Uma alternativa citada pelo autor é a utilização de regras de associação que, por tentar identificar padrões na base de dados, acabam não sendo tão afetadas por amostras pequenas de ruído. Essas regras, depois de extraídas, poderiam ser aplicadas como heurística no passo intermediário, como comentado previamente.

A esparsidade de uma base de dados é representada pela quantidade de itens não avaliados, ou seja, as células em branco na matriz Usuário X Item. Quanto mais alta a esparsidade, maior também é a dificuldade de encontrar usuários similares e de consequentemente gerar recomendações utilizando a FC.

4.4.2.3. Filtragem Baseada em Informação Contextual

Adomavicius e Tuzhilin (2011) afirmam que contexto é um conceito estudado por diferentes disciplinas como Ciência da Computação, Filosofia, Psicologia, etc. Cada área tende a construir sua própria visão, que é normalmente um pouco diferente das outras.

A fim de trazer uma visão mais genérica, Dourish (2004) introduz a taxonomia de contextos, classificando-a em duas visões: representacional e interacional. A primeira visão, representacional, define contexto como um conjunto de atributos observáveis, possíveis de serem descobertos *a priori* e que não variam significativamente sua estrutura no tempo. Em contraste, a visão interacional assume que o comportamento do usuário é induzido por certo contexto, não sendo necessariamente observável.

Focando em Sistemas de Recomendação e nas áreas de pesquisa relacionadas, vários atributos podem ser utilizados para compor um contexto, como por exemplo, a intenção ao realizar uma compra, localização, identidade de pessoas próximas, objetos ao redor, estação do ano, temperatura, status emocional, personalidade, tempo e companhia do usuário. O contexto também pode ser aplicado ao usuário ou à aplicação.

A informação contextual, devido a sua complexidade, pode ser composta por uma estrutura hierárquica possível de ser representada em formato de árvores (Adomavicius e Tuzhilin 2011). Um exemplo de hierarquia de contexto relacionado a programa televisivo: Programa: Programa □ Subgênero □ Gênero □ Canal; Tempo: Hora □ Data □ Dia da Semana □ Mês □ Ano.

Em vários domínios, como por exemplo recomendar um pacote de férias, um conteúdo personalizado em um site *Web*, produtos em uma loja *online* ou um filme, seria interessante

incorporar informação contextual no processo de recomendação, tais como usuários, itens, tempo e lugar. De acordo com Giordani (2006), a criação de contextos de interesse aumenta as possibilidades de definir o momento mais adequado para recomendar certo produto e, principalmente, quais os produtos que devam compor o perfil do usuário em determinado contexto.

As técnicas de filtragem estudadas aqui utilizam uma função R (Equação 1) e se baseiam exclusivamente nas variáveis usuário e item.

$$R: \text{Usuário} \times \text{Item} \rightarrow \text{Avaliações} \quad (1)$$

A Filtragem Baseada em Informações Contextuais incorpora informação contextual no processo de recomendação, dentro do cenário de decisão do usuário, tais como usuários, itens, tempo e lugar, criando, desta forma, uma função R (Equação 2) com a utilização do contexto:

$$R: \text{Usuário} \times \text{Item} \times \text{Contexto} \rightarrow \text{Avaliações} \quad (2)$$

Alguns tipos de informação contextual podem ser mais relevantes para certo tipo de aplicação do que para outros. Para determinar a relevância de certa informação, um especialista no negócio ou um arquiteto de sistemas de recomendação pode manualmente definir a importância, como também utilizar recursos de aprendizado de máquina, mineração de dados e estatísticas sobre as avaliações já existentes. Segundo Giordani (2006), a construção de contextos individuais não é tão simples pois o analista de negócios deveria efetuar uma análise dos dados transacionais de cada cliente em particular, para então tentar identificar possíveis contextos. Neste caso, fica clara a inviabilidade das análises manuais em função da massificação dos dados, tornando necessária a utilização de artifícios que permitam uma descoberta automatizada computacionalmente.

4.4.2.4. Filtragem Baseada em Outros Contextos

Tkalčič et. al. (2009) afirmam que no contexto de TVD, Sistemas de Recomendação que ignoram as experiências afetivas do usuário enquanto o mesmo está consumindo conteúdo multimídia são no mínimo estranhos, devido ao fato de a indústria do entretenimento estar baseada em causar emoções nas pessoas.

De acordo com a visão de McDonald (2003) a mudança mais importante a se desenvolver na nova geração de Sistemas de Recomendação é a devida complexidade na construção do modelo/perfil de usuário e, o uso apropriado desse modelo. Considerando Perugini et al (2004) modelos/perfis de usuário propiciam indiretamente conexões entre pessoas possibilitando e direcionando a recomendações mais eficientes. Dessa forma, acredita-se que perfis de usuário devem representar diferentes e ricos aspectos da experiência diária de um usuário, considerando a vida real como modelo.

Considere o percurso de um sistema de computador para atingir a mínima compreensão desta otimizada interação de como os humanos procedem nos seus processos de recomendação na “vida real”. Note que humanos usam em suas recomendações informações mais complexas que informações efetivamente usadas por um computador. Isto é, usualmente, sistemas computacionais usam informações tais como competências, preferências, informações demográficas dos usuários, entre outras, para a tomada de decisão e posterior recomendação de uma informação, produto ou serviço. Em contraste, humanos, quando tomam decisões, recomendam e personalizam informações, produtos e serviços para outros humanos, além de usar as informações convencionais usadas pelos sistemas computacionais tendem, também, a usar informações adicionais relacionadas

a habilidades sociais e psicológicas humanas, tais como, traços de personalidade e Emoção (Nunes and Aranha 2009).

Mesmo sabendo ser impossível perfeitamente antecipar as necessidades humanas individuais para recomendar o produto certo, sabe-se que quanto mais ricas forem as informações sobre o usuário, mais precisos serão os produtos, serviços e /ou pessoas recomendadas. Basta observar o que ocorre na vida real.

Nessa linha, note que Nunes (Nunes 2009) expande o modelo inicial de técnicas de recomendação proposto por Gonzalez et al (2007). Gonzalez por sua vez expande o modelo proposto por Burke (2002). O modelo de Burke (2002) categoriza as técnicas de recomendação em cinco: baseada em conteúdo, filtragem colaborativa, demográfica, baseada em conhecimento e baseada em utilidade. Gonzalez et al (2007) por sua vez cria uma nova categoria intitulada recomendação baseada em outros contextos, incluindo aspectos Psicológicos, tais como, Inteligência Emocional e Interação Social, usando como fonte um Perfil de usuário baseado em contextos diferenciados dos tradicionais modelos demográficos ou baseados em preferência do usuário.

Considerando esse aspecto Nunes (2009) ao expandir o modelo proposto por Gonzalez et al (2007), apresenta a personalidade do usuário como um fator relevante na otimização das recomendações ao usuário, como apresentado na Figura 3.

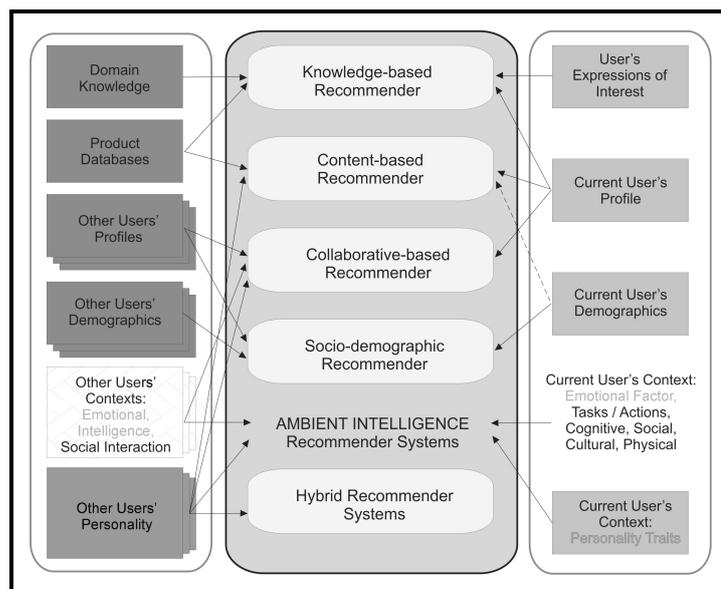


Figura 3. Sistemas de Recomendação baseado em personalidade (Nunes 2009)

A técnica de recomendação baseada em personalidade geralmente é aplicada juntamente a outra técnica como a filtragem colaborativa e/ou baseada em conteúdo, representando uma técnica de recomendação híbrida.

Note que para que seja viável a recomendação baseada na personalidade do usuário necessita-se a representação da personalidade, cada projetista de sistema pode usar uma abordagem diferenciada.

Cazella et al (2010) afirmam que a técnica baseada em personalidade geralmente é aplicada juntamente a outra técnica como a filtragem colaborativa e/ou baseada em conteúdo, constituindo-se numa técnica de recomendação híbrida.

4.4.2. 5. Filtragem Híbrida

A filtragem híbrida busca unir múltiplas técnicas de recomendação a fim de aproveitar o melhor delas (Sandvig et al 2007). É possível, ainda, combinar duas diferentes técnicas do mesmo tipo, como por exemplo, duas diferentes técnicas baseadas em conteúdo. Uma outra alternativa, é manter os perfis dos usuários baseados em uma análise de conteúdo, comparando-os diretamente, por meio de uma recomendação colaborativa, para determinar possíveis semelhanças (Herlocker 2000; Ansari 2000).

Segundo Herlocker (2000), todas as técnicas baseadas em conhecimento (baseada em conhecimento, baseada em conteúdo, colaborativa e demográfica) sofrem com o problema do *cold start* de uma forma ou outra, seja para tratar de novos itens ou novos usuários. Porém, a junção de técnicas de tipos diferentes é indicada para tentar resolver este problema (Sandvig et al 2007).

Com o uso da filtragem híbrida e utilizando os benefícios da filtragem colaborativa e da filtragem baseada em conteúdo, consegue-se a descoberta de relacionamentos entre usuários, a recomendação de itens baseado na avaliação dos mesmos, o tratamento de usuários incomuns e a precisão independente do número de usuários e avaliações (Cazella 2006).

4.4.3. Algoritmos de recomendação

A seguir serão descritos brevemente dois algoritmos aplicados a sistemas de recomendação.

4.4.3. 1. Algoritmo para Filtragem baseada em conteúdo

A Filtragem Baseada em Conteúdo parte do princípio de que os usuários tendem a interessar-se por itens similares aos que demonstraram interesse no passado. Desta maneira, é definida a similaridade entre os itens. Em alguns casos, pode haver maior dificuldade para estabelecer esta similaridade (Adomavicius et al 2005). Para que seja estabelecida a similaridade entre itens como roupas e brinquedos, por exemplo, seria necessária a identificação dos atributos nos itens a serem comparados (peso, preço, marca, por exemplo). No caso dos itens serem artigos científicos (ou documentos), o processo de comparação pode ser focado nos termos que compõem estes textos (Selton e McGill 1983).

Para que seja estabelecida a similaridade entre artigos (ou documentos), este processo de comparação pode ser facilitado, pois documentos podem ser considerados similares se compartilharem termos em comum. Sendo assim, a filtragem baseada em conteúdo é mais indicada para a recomendação de itens textuais, onde o conteúdo é geralmente descrito com *keywords* (Selton e McGill 1983). Um exemplo é o sistema de recomendação Fab, o qual recomendava páginas *web* para usuários. Este sistema representa uma página *web* com os seus 100 mais importantes termos. Similarmente, o sistema Syskill & Webert representa os documentos com as 128 palavras mais significativas (Adomavicius et al 2005).

Outros exemplos de filtragem baseada em conteúdo são sistemas booleanos de recuperação, onde a consulta constitui-se um conjunto de palavras-chave combinadas com operadores booleanos; sistemas de filtragem probabilística, onde o raciocínio probabilístico é utilizado para determinar a probabilidade que um documento possui de atender as necessidades de informação de um usuário; e linguagem natural (Herlocker 2000).

Conforme mencionado anteriormente sistemas de recomendação baseados em conteúdo podem recomendar itens similares a itens que o usuário gostou no passado. Deste modo, vários itens são comparados com itens que foram avaliados positivamente e os mais similares serão recomendados ao usuário. Conforme apresentado por Adomavicius et. al. (2005) pode-se formalizar esta definição estabelecendo *ContentBasedProfile(c)* como sendo o perfil do usuário *c*. Este perfil é

obtido através de uma análise do conteúdo dos itens previamente avaliados pelo usuário utilizando técnicas de recuperação de informação. Por exemplo, $ContentBasedProfile(c)$ pode ser definido como um vetor de pesos (wc_1, \dots, wc_k) onde cada peso wc_i denota a importância do termo k_i para o usuário c utilizando-se a medida TF-IDF (*term frequency–inverse document frequency*).

Em sistemas de recomendação baseados em conteúdo, a função utilidade $u(c,s)$ é geralmente definida conforme a equação (3):

$$u(c,s) = score(ContentBasedProfile(c), Content(s)) \quad (3)$$

Tanto o $ContentBasedProfile(c)$ do usuário c como o $Content(s)$ podem ser representados como vetores (TF-IDF) de pesos e termos \vec{w}_c e \vec{w}_s . Além disso, a função utilidade $u(c,s)$ normalmente é representada, na literatura de recuperação de informação, por algum tipo de pontuação heurística sobre vetores, como por exemplo, a medida de similaridade do cosseno. O cálculo para a medida de similaridade do cosseno pode ser feito de acordo com a equação (4) (Adomavicius et al, 2005):

$$u(c,s) = \cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{\|\vec{w}_c\|_2 \times \|\vec{w}_s\|_2} = \frac{\sum_{i=1}^k w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^k w_{i,c}^2} \sqrt{\sum_{i=1}^k w_{i,s}^2}} \quad (4)$$

Onde k é o número total de palavras no sistema. Desta forma, o cálculo de similaridade é realizado computando o cosseno do ângulo formado pelos dois vetores que representam os documentos (termos e frequências) (Adomavicius et al 2005). Esta abordagem é utilizada no protótipo desenvolvido para calcular a similaridade entre os artigos. Por exemplo, se o usuário c , lê muitos artigos relacionados ao tópico Engenharia de Software, as técnicas de filtragem de conteúdo estão aptas a recomendar outros artigos de Engenharia de Software para o usuário c . Estes outros artigos são recomendados porque possuem termos em comum, que neste exemplo, são termos do domínio Engenharia de Software. Neste caso, a função $u(c,s)$ será alta, pois os pesos calculados para os termos destes artigos (que compartilham termos em comum referente ao assunto Engenharia de Software) serão altos (Adomavicius et al 2005).

4.4.3. 2. Algoritmo para Filtragem Colaborativa

A técnica de filtragem colaborativa (FC) possui um modelo conceitual de operação de fácil entendimento, possibilitando analisar itens a serem recomendados sem preocupar-se com o conteúdo destes itens e sim focando nas avaliações dos itens. A operação de um sistema de recomendação por filtragem colaborativa é similar a recomendação verbal de pessoa para pessoa. Os usuários são supridos de recomendação seguindo três etapas:

- 1) usuário fornece seu perfil de avaliações;
- 2) a FC identifica usuários com perfis similares (vizinhos);
- 3) as avaliações dos vizinhos são combinadas para se gerar as recomendações. Com isso, a percepção do usuário dos passos, relacionados anteriormente, afetará sua percepção sobre todo o sistema.

A seguir as etapas serão detalhadas: na Etapa 1, calcula-se o peso em relação a similaridade do usuário-alvo (aquele que deverá receber uma recomendação) e os demais usuários. Isso pode ser calculado aplicando, por exemplo, o coeficiente do Co-seno ou o coeficiente de Pearson (Cazella, et al 2010); na Etapa 2, seleciona-se, a partir dos resultados do cálculo da similaridade, um subconjunto de usuários com maiores níveis de similaridade, que serão denominados vizinhos. Estes vizinhos serão considerados no cálculo da predição do item a ser recomendado; na Etapa 3, normaliza-se as avaliações fornecidas pelos usuários ao item em análise para recomendação e

calcula-se a predição, ponderando-se as avaliações dos vizinhos com seus respectivos pesos de similaridade.

A Tabela 1 apresenta na prática como a filtragem colaborativa funciona. Por exemplo, se quisermos recomendar um produto ao usuário Mauro, procuraremos outros usuários com hábitos de consumo semelhantes. No caso, Paulo e João já compraram produtos que Mauro também comprou (*Prod2*). Em seguida, recomendamos a Mauro produtos que estes dois outros usuários possuem, mas que Mauro ainda não possui, como *Prod1* e *Prod5*. A decisão sobre a recomendação destes produtos baseia-se no histórico de avaliações comuns e o valor de predição calculado.

Tabela 1. Recomendação baseada em filtragem colaborativa

Usuário	Prod ₁	Prod ₂	Prod ₃	Prod ₄	Prod ₅	Prod ₆
Paulo		x			X	
João	x	x				
Márcia			x	x	X	
Carlos			x			
Ana	x			x		
Mauro	?	x			?	

4.4.3. 2. 1. Cálculo do Coeficiente de Similaridade

Para o cálculo da similaridade pode-se adotar o coeficiente de Pearson, uma vez que o mesmo é amplamente utilizado em modelos e mede o grau de relacionamento entre duas variáveis, variando de -1 (ausência total de correlação) a 1 (forte correlação). O cálculo deste coeficiente é realizado conforme a equação 5 (Cazella et al 2010):

$$corr_{ab} = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2 \sum_i (r_{bi} - \bar{r}_b)^2}} \quad (5)$$

Sendo o $corr_{ab}$ a correlação do usuário alvo a com um determinado usuário b ; r_{ai} : é a avaliação que o usuário ativo a atribuiu para o item i ; r_{bi} : é a avaliação que o usuário ativo b atribuiu para o item i ; \bar{r}_a é a média de todas as avaliações do usuário ativo a , em comum com o usuário b ; \bar{r}_b é a média de todas as avaliações do usuário ativo b , em comum com o usuário a .

4.4.3. 2.2. Predição

A predição é feita independentemente do coeficiente utilizado no cálculo da similaridade, pois ela será gerada através de uma média ponderada das avaliações dos vizinhos que obtiveram um coeficiente de similaridade aceitável, ou seja, com limiar igual ou superior, por exemplo, a 0,3. A equação 6 é utilizada para o cálculo da predição (Cazella et al 2010).

$$p_{ai} = \bar{r}_a + \frac{\sum_{b=1}^n (r_{bi} - \bar{r}_b) * corr_{ab}}{\sum_{b=1}^n |corr_{ab}|} \quad (6)$$

Sendo $corr_{ab}$ é a correlação do usuário alvo a com um determinado usuário b ; p_{ai} a predição de um item i para um usuário alvo a ; \bar{r}_a é a média de todas as avaliações do usuário alvo a aos itens que foram pontuados por todos os seus usuários similares; r_{bi} é a avaliação que o usuário ativo b atribuiu para o item i ; \bar{r}_b é a média de todas as avaliações do usuário b , em comum com o usuário a .

4.5. Exemplos práticos de uso de personalidade para tomada de decisão computacional

Considerando que os estudos realizados nesse capítulo demonstram uma nova tendência à renovação dos Sistemas de Recomendação, onde os aspectos psico-afetivos humanos devem ser considerados na tomada de decisão computacional, aqui apresenta-se alguns exemplos.

4.5.1. e-commerce, e-services

4.5.1.1. Hu and Pu (2009a)

Segundo a visão de Pina e Nunes (2011) o trabalho relata um estudo realizado com usuários para elicitare as preferências entre duas abordagens de sistemas de recomendação: baseado em classificação e baseado em teste de personalidade.

Hu e Pu argumentam que tecnologias de recomendação foram incorporadas a uma série de aplicações, especialmente para sites de *e-commerce* como ferramentas poderosas para ajudar a lidar com a sobrecarga de informações de usuários e prestação de serviços personalizados. Diante desta constatação Hu e Pu realizaram um experimento para comparar dois sites de recomendação de filmes. Para tanto foram adotados os sistemas : *MovieLens*³ e *WhattoRent*⁴.

Para execução do experimento foram feitas algumas adaptações na interface do *MovieLens* para que as diferenças existentes entre ele e o *whattorent* não afetasse a satisfação dos usuários no sistemas. Foi adicionado um link atrelado a cada filme recomendado que exibe um pequeno o clipe do filme. O objetivo é de ajudar o usuário no processo de avaliação dos filmes recomendados. Foi também inserido um grupo de botões ao lado de cada filme recomendado para facilitar aos usuários o fornecimento das avaliações em cinco níveis que vão de muito desinteressada até muito interessado.

O experimento foi feito com 30 participantes, sendo 11 mulheres e 19 homens, distribuídos em quatro grupos etários e com diferentes profissões (cada usuário recebeu umauxilio financeiro como motivação). Cada um dos participantes avaliou os dois sistemas de recomendação (*MovieLens* e *WhattoRent*).

A execução do experimento foi no gabinete de um administrador que supervisionava a experiência e ajudava os participantes a concluir todas as tarefas com sucesso. A seguir são listados os passos do experimento: (i) Participante registra uma nova conta para evitar quaisquer influências a partir do histórico de uso; (ii) No *MovieLens* o participante classifica 15 filmes que tenha visto antes e no *whattorent* o participante responde a 20 questões de personalidade; (iii) São respondidas duas perguntas sobre o atual humor do usuário; (iv) O participante classifica 6 filmes recomendados em uma escalas que vai de 1 = muito desinteressado a 5 = muito interessado; (v) O participante responde um questionário de avaliação on-line onde cada pergunta é respondida em uma escala *Likert* sobre as preferências entre os dois sistemas e suas razões.

Os critérios de avaliação utilizados foram: (i) Percepção de Precisão - Mede quanto os usuários sentem que as recomendações correspondem às suas preferências; (ii) Esforço do Usuário - Mede a

³ O *MovieLens* é um sistema de recomendação baseado em classificação, foi construído pelo grupo de pesquisa GroupLens liderado por John Riedl da Universidade de Minnesota. O *MovieLens* aprende as preferências dos usuários a partir de suas avaliações e utiliza a tecnologia de filtragem colaborativa para recomendar filmes. Quando um novo usuário faz logon, ele é obrigado a informar pelo menos 15 filmes que já tinha visto. Estas classificações são usadas para construir o perfil de usuário e encontrar os ‘vizinhos’ que têm gostos similares ao do presente usuário. Com base na classificação a partir desses vizinhos, ele prediz quanto o usuário possivelmente irá gostar de um filme. Em seguida, ele apresenta todas as recomendações de Filmes em ordem decrescente dos escores previsto. Usuários podem refinar os seus perfis de preferência classificando mais filmes e também podem alterar as classificações anteriores em interações futuras com o sistema.

⁴ O *whattorent* é um sistema de recomendação baseado em personalidade, foi construído com base na Teoria *LaBarrie* que afirma que os “telespectadores interagem emocionalmente com um filme da mesma maneira que eles interagem com outros seres humanos”. Seus desenvolvedores elaboraram um questionário com 20 questões de personalidade onde as preferências dos usuários são decifradas e é feita a recomendação do filme.

quantidade de esforço subjetivo que os usuários sentiram que gastaram e o tempo real que foi utilizado para completar processo de preferência; (iii) Fidelidade do Usuário- Avalia se o sistema tem a capacidade de convencer seus usuários a adquirir o produto recomendado, se o sistema gera interesse de reutilização ou o sistema é satisfatório ao ponto do participante fazer uma indicação para os seus amigos; (iv) Os resultados mostram que a percepção da precisão entre os dois sistemas não é significativamente diferente. No entanto, os usuários gastam muito menos esforço para completar o perfil de preferência baseado em personalidade do que no sistema baseado em classificação. Além disso, os usuários expressaram forte intenção de reutilizar o sistema baseado em personalidade e apresentá-lo aos seus amigos. Após analisar os resultados do questionário final sobre preferência, foi possível constatar que 53,33% dos usuários (16 em 30) preferiram o sistema baseado em teste de personalidade, enquanto que 13,33% dos usuários (4 em 30) preferiram o sistema baseado em classificação.

As autoras acreditam que a principal contribuição do artigo foi perceber quanto a extração de preferências baseada em abordagens de personalidade pode ajudar aos usuários a revelar as preferências ocultas e assim aumentar a precisão recomendação. Ainda o artigo demonstrou a preferência dos usuários por sistemas de recomendação baseado em personalidade. Uma das razões que justificaram essa preferência foi a redução do esforço. Porém os usuários que preferiram a recomendação convencional justificaram a sua preferência considerando a vantagem importante de ter controle sobre a criação do seu próprio perfil.

4.5.1.2. *Hu and Pu (2009b)*

Segundo Pina e Nunes (2011) esse trabalho complementa o trabalho apresentado na seção anterior. O artigo avalia a aceitação de usuários aos sistemas de recomendação baseado em personalidade (RBP), usando o modelo de aceitação de tecnologia (TAM).

O artigo utiliza o mesmo experimento da seção anterior porém utiliza uma formalização melhor para referenciar os resultados. O experimento, como descrito anteriormente foi realizado com 30 participantes, sendo que cada um dos participantes avaliou os dois sistemas de recomendação (*MovieLens* e *WhattoRent*).

A avaliação usou os critérios baseados no modelo TAM, que foi projetado para compreender a relação causal entre variáveis externas de aceitação dos usuários e o uso real do computador, buscando entender o comportamento deste usuário através do conhecimento da utilidade e da facilidade de utilização percebida por ele (DAVIS et al 1989). Segundo Davis et al (1989) as pessoas tendem a usar ou não uma tecnologia com o objetivo de melhorar seu desempenho no trabalho, esse fator é chamado de utilidade percebida. Porém, mesmo que essa pessoa entenda que uma determinada tecnologia é útil, sua utilização poderá ser prejudicada se o uso for muito complicado, de modo que o esforço não compense o uso, esse fator é chamado de facilidade percebida. É também avaliado o critério de intenções comportamentais para uso. Onde são avaliados três aspectos: intenção de compra, a intenção de voltar ao sistema e a intenção de recomendar este sistema para amigos.

Em relação a utilidade percebida - os filmes recomendados pelo sistema baseados em personalidade foram avaliados como de interesse 54,9%, comparados àqueles recomendados pelo sistema baseado em classificação com 42,7% de interesse. Em relação a facilidade de uso - o esforço global cognitivo percebido é significativamente menor no sistema baseado em personalidade do que no sistema baseado em classificação. Em relação as intenções comportamentais de uso – nos três aspectos os RBP obteve um vantagem. Os resultado final encontrado da comparação de aceitação entre os dois sistemas de recomendação mostram que

53,33% dos usuários preferiram o sistema baseado em personalidade em relação a 13,33% do usuários que preferiram o sistema baseado em classificação.

Assim o artigo permitiu concluir a preferência dos usuários entre sistemas de recomendação baseado em personalidade e sistema de recomendação baseado em classificação. E também revelou percepções sobre o processo de escolha do usuário, mostrando que um design de interface simples, pouco esforço inicial, e recomendação de boa qualidade são os fatores mais influentes que contribuem para a aprovação dos usuários nos sistemas de recomendação.

A pesquisa mostrou que a personalidade é um fator de resistência primária e que determina os comportamentos humanos e que são significativas as ligações entre a personalidade e os interesses das pessoas. Sistemas de recomendação baseados em personalidade podem fornecer informações e serviços mais personalizados, desde que eles compreendam melhor os clientes sob o ponto de vista psicológico.

4.5.1.3 Recio-Garcia et al (2009)

Segundo Pina e Nunes (2011) o artigo apresenta um novo método de realizar recomendações para grupos usando a técnica de filtragem colaborativa e considerando a composição da personalidade do grupo. A abordagem utiliza personalidades de membros do grupo para escolher o filme mais interessante que pode melhor satisfazer todo o grupo. A partir dos algoritmos de recomendação clássicos foram feitas três adaptações diferentes utilizando como base o Peso Modo Conflito (CMW) aplicado aos membros do grupo. A pesquisa foi testada no domínio de recomendação de filmes, usando os dados *MovieLens* em conjunto com grupos de usuários de diferentes tamanhos e graus de homogeneidade.

Os autores relatam que conflito é uma parte natural de nossas interações com os outros. Diferentes pessoas têm diferentes expectativas e desejos que geralmente parecem ser incompatíveis. Em situações de conflito, pode-se descrever o comportamento de um indivíduo sob duas dimensões básicas: (i) Assertividade - medida em que a pessoa tenta satisfazer suas próprias preocupações; (ii) Cooperativismo - medida em que a pessoa tenta satisfazer as preocupações das outras pessoas.

O teste TKI (Instrumento Modo Conflito Thomas-Kilmann) foi escolhido pois se concentra na gestão de conflito. O TKI constrói um perfil do usuário por meio de 30 perguntas de escolha única. O teste fornece pontuações para enquadrar o usuário dentre os modos: Competitivo, Colaborativo, Acomodado, Comprometido e Cauteloso. Os cinco modos representam as preferências do indivíduo quando têm que enfrentar situações de conflitos. Estas pontuações são normalizadas utilizando uma amostra de 8.000 pessoas. O Valor do Peso Modo Conflito (CMWU) representa o comportamento do indivíduo após a avaliação TKI. É calculado usando a seguinte equação: $CMW = 1 + \text{Assertivo} - \text{Cooperativismo}$. A seguir a Tabela 2 demonstra a escala de pontuação dos cinco modos utilizada para extrair o nível assertivo ou cooperativo do indivíduo para serem utilizados na equação para encontrar o CMWU:

Tabela 2. Tabela dos coeficientes para calcular CMW Fonte: (Recio-Garcia et al 2009)

TKI Mode	Assertiveness		Cooperativeness	
	High	Low	High	Low
Competing	0.375	-0.075	-0.15	0
Collaborating	0.375	-0.075	0.375	-0.075
Compromising	0	0	0	0
Avoiding	-0.375	0.075	-0.375	0.075
Accommodating	-0.15	0	0.375	-0.075

No contexto de recomendações para grupos as principais abordagens para gerar preferência de grupos com base nas preferências de um usuário são: (i) Fusão das recomendações feitas para os

indivíduos (união ou interseção); (ii) Agregação de indivíduos para classificação; (iii) Construção de um modelo de preferência do grupo. Nesse estudo foi escolhida a abordagem (ii) para recomendação de grupo onde se assume que um indivíduo fornece previsões de classificação para cada usuário do grupo e para cada item a ser recomendado.

Foram implementados três algoritmos de recomendação de grupo a partir do algoritmo básico para que incluíssem o CMW de uma forma diferente. Posteriormente foram feitas comparações entre os resultados das três versões. Os algoritmos são: (i) Minimizando penalização - Este algoritmo utiliza o CMW para calcular o quão interessante é o melhor conjunto de itens candidatos da recomendação de um membro para as outras pessoas do grupo; (ii) Média de satisfação - Este algoritmo é inspirado no procedimento médio de satisfação, o critério de seleção escolhe n itens com maior média de satisfação; (iii) Menor sofrimento - Este algoritmo propõe representar uma variação do procedimento de minimização do sofrimento empregados pelas classificações inferida pelo *MovieLens* para gerar a agregação de preferências.

Os autores apresentaram um experimento com um conjunto de 70 alunos no domínio de recomendação de filmes. Os alunos preencheram suas preferências sobre os filmes dentre uma lista com 50 filmes heterogêneos selecionados a partir do conjunto de dados do *MovieLens*. Em média os usuários avaliaram 33 filmes. Em um outro momento os alunos fizeram o teste TKI, possibilitando o cálculo do Valor Peso Modo Conflito (CMW) para cada indivíduo. Na seqüência o CMW é aplicado nos três algoritmos já citados para obter as recomendações para os grupos.

Para avaliar os resultados foi feita uma comparação usando uma medida simples de avaliação que conta quantos filmes classificados pelo usuário estão no conjunto dos melhores filmes sugeridos pelo sistemas de recomendação. Ao analisar os resultados dos algoritmos com a aplicação do fator CMW foi possível constatar que o algoritmo minimização penalizações funciona melhor propondo apenas um filme. Com a adoção do algoritmo média de satisfação e menor sofrimento é refletida uma melhoria geral na precisão da recomendação. Os resultados também melhoram quando aplicados em grupos com pessoas que tenham soluções de conflito de personalidades heterogêneas.

O artigo contribui demonstrando que sistemas de recomendação para grupos poderiam ser melhorados em até 7% de precisão ao utilizar os valores da personalidade dos indivíduos obtidos a partir do TKI e CMW.

4.5.2. TV digital (Trevisan 2011)

Uma aplicação para recomendação de programas de TV (Trevisan 2011) pode rodar em diferentes aparelhos eletrônicos: celulares, computadores e televisores, entre outros com algum tipo de acesso a rede mundial de computadores. Porém, o mais usual é disponibilizar este tipo de aplicação em um aparelho de televisão. Com o surgimento da TV Digital no Brasil e a crescente expansão do sinal, a possibilidade do desenvolvimento de programas interativos tornou acessível a criação deste tipo de sistema. Então, utilizando o SBTVD, foram testadas as duas plataformas do Ginga (Ginga NCL e Ginga-J) para o desenvolvimento do protótipo da aplicação. O Ginga NCL foi o que apresentou maior maturidade e que possui o maior número de aplicações sendo desenvolvidas no cenário nacional, até por conta da maior maturidade que possui. Os testes foram realizados no emulador XletView que utiliza uma implementação de referência do MHP (*middleware* do padrão europeu). O XletView é um emulador de TVD baseado no *middleware* MHP que permite a execução de xlets em um ambiente PC. É um projeto de *software* livre e está sob a licença GNU GPL. Um xlet é uma aplicação Java para TVD e equivale a um applet Java em um PC (Araujo e Carvalho 2009).

O protótipo em TVD implementado acessa um servidor HTTP e busca as recomendações disponíveis para o usuário que o está utilizando. O telespectador pode identificar o quanto o

programa recomendado atende às suas necessidades ou desejos. Esta informação alimenta o sistema como uma nova avaliação, aumentando a precisão de futuras sugestões. A base de dados de programas de televisão foi disponibilizada pela empresa Revista Eletrônica para o presente trabalho e pôde ser obtida via FTP, no formato XML. A grade de programação utilizada no protótipo foi extraída entre 21/06/2010 a 05/07/2010 (quinze dias). A fim de viabilizar o experimento e devido à quantidade de usuários e o tempo de experimentação, foi decidido pela importação de somente programas dos canais abertos da televisão brasileira: o SBT, a RBS, a Bandeirantes, a TV COM, a TV Futura e a Rede TV. Desta forma, foi obtida uma base de dados com uma esparcialidade não tão alta e com programas mais conhecidos.

O contexto cujo usuário está inserido no momento da recomendação é informado pelo mesmo de forma explícita a partir de uma lista pré-definida, porém configurável. Estes contextos foram elencados empiricamente a partir da análise dos gêneros, subgêneros e classificação etária dos programas disponíveis na grade de programação. Como os contextos são configuráveis, é possível a aplicação de um algoritmo de mineração de dados no momento em que há a posse de uma base de dados com uma quantidade maior de avaliações e extrair novos contextos, talvez mais significativos. Os contextos elencados para a execução do protótipo foram por exemplo, 1) Nome do contexto: Sozinho; Descrição: Adulto assistindo sozinho; Aparelho: Televisor; 2) Nome do contexto: Com Crianças; Descrição: Acompanhado de crianças; Aparelho: Televisor; 3) Nome do contexto: Recebendo Amigos; Descrição: Recebendo amigos para um jantar; Aparelho: Televisor; 4) Nome do contexto: Com Namorado(a); Descrição: Acompanhado do namorado(a)/esposo(a); Aparelho: Televisor; 5) Nome do contexto: Locomovendo-se; Descrição: Viajando de ônibus para o trabalho; Aparelho: Celular.

A biblioteca Apache Mahout foi utilizada como *framework* para a construção das recomendações. Além de vários outros recursos, ela fornece algoritmos de filtragem colaborativa e baseada em conteúdo, estando sob a licença de software da Apache (Mahout, 2011). Os testes de personalidade foram realizados utilizando a aplicação *Personality Inventory PVI.0* Nunes et al (2010). Esta é uma aplicação *online* que utiliza os testes NEO-IPIP e TIPI. Cada questão respondida tem um valor atribuído entre 1 e 5 (Figura 1c). A partir destes valores, é realizado um cálculo que gera um valor entre 1 e 100 para cada um dos itens do Big Five.

A seleção do contexto em que o usuário se encontra é realizada de forma explícita pelo mesmo antes de receber uma recomendação. Pode ser alterada a qualquer momento, pressionando a tecla verde do controle remoto. Quando a lista é exibida, basta clicar com a tecla de comando “para cima” e “para baixo” do controle remoto até posicionar sobre o contexto desejado e, então, pressionar o botão “ok” do mesmo, conforme a Figura 4.

Uma vez definido o contexto do usuário, o sistema acessa o servidor e busca uma lista de recomendação para o usuário. É possível que nenhum programa seja recomendado para o usuário, neste momento, pelo algoritmo descrito no modelo do sistema anteriormente. Para que o usuário não fique sem receber nenhuma recomendação, o sistema busca por programas que se enquadrem no contexto selecionado e que possua os gêneros, subgêneros e classificações etárias melhor avaliadas por este usuário no passado.

Neste ponto, o sistema acaba realizando uma filtragem baseada em conteúdo e, desta forma, o usuário não fica frustrado por não receber nenhum item recomendado e acaba com o problema do *cold start*. Outro ponto positivo de recomendar itens mesmo que o algoritmo de filtragem colaborativa não retorne nenhuma recomendação é a alimentação do sistema com o *feedback* do usuário que é enviado para cada recomendação gerada, o que de certa forma contribui para o sucesso das suas próximas utilizações.

A lista de itens recomendados apresenta o título do programa, o ícone do canal, a classificação etária e a avaliação dada pelo usuário representada pelas estrelas ao lado de cada programa (Figura 5). Para navegar pelos itens e avaliá-los, basta que o usuário utilize as teclas de navegação do controle remoto “para cima” e “para baixo” a fim de posicionar-se no item desejado, e a tecla “para esquerda” e “para direita” para aumentar ou diminuir a nota dada para o programa recomendado.

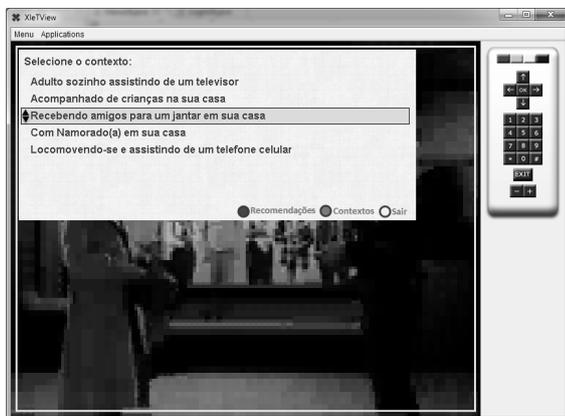


Figura 4 – Seleção explícita de contexto (Trevisan 2011).

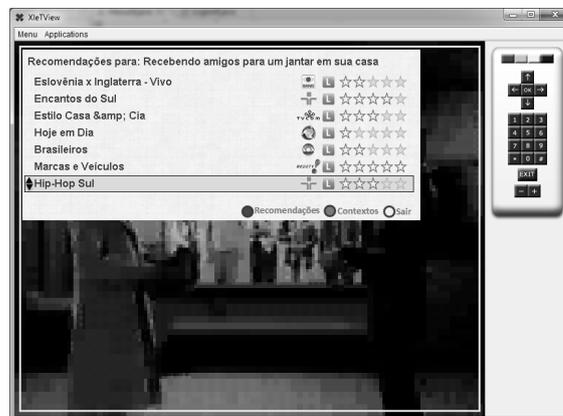


Figura 5 – Lista de recomendações fornecidas pelo TvPlus (Trevisan 2011).

Cada avaliação realizada pelo usuário pela TV é enviada para o servidor e armazenada na tabela de avaliações. Quando o algoritmo de recomendação for executado novamente por qualquer usuário do sistema, estas informações transmitidas no *feedback* realizado serão processadas e poderão gerar uma recomendação diferente das anteriormente criadas.

4.5.3.Outros

4.5.3.1.Presidencial Francesa (Nunes 2009)

Nunes (2009) apresenta um protótipo de Sistema de Recomendação que objetiva dar indícios do quão importante os aspectos psicológicos, como traços de personalidade, são no processo de tomada de decisão computacional, isto é, no processo de recomendação (em Sistemas de Recomendação e/ou Sistemas de Combinação Social).

A Presidencial Francesa é um Sistema de Recomendação mostrando sua habilidade para recomendar pessoas, nesse caso, consideradas no contexto de produtos e não prestadoras de serviço⁵.

Cenário: O cenário desse sistema é apresentado pelas “eleições Presidenciais na França” realizada em 2007. Nesse caso, o Sistema de Recomendação foi usado para gerar uma recomendação considerando a melhor escolha de um candidato para uma pessoa votar, considerando a reputação dos candidato sob as vistas da própria pessoa que receberia a recomendação. Esse experimento foi aplicado de dezembro de 2006 a julho de 2007.

⁵ O Sistema de Recomendação gera uma pessoa com um “produto”, pois, nesse caso, a pessoa é considerada como um pacote fechado, ou seja, a pessoa significa verdadeiramente o nome dessa pessoa. Nesse caso, a pessoa não é considerada como um provedor de serviço, como normalmente as pessoas são consideradas. A visão de produto significa que a pessoa recebe uma recomendação de alguém constando um nome a ser considerado como suporte a sua tomada de decisão, diferentemente da visão de serviço. Na visão de serviço, a pessoa recebe um nome para ser usado como um provedor de serviço, isto é, este “nome”(pessoa) irá provavelmente executar algum serviço a posteriori. Na visão de produto, o Sistema de Recomendação gera uma resposta passiva, diferente da visão de serviço onde a resposta, provavelmente, gera uma interação dinâmica com a pessoa que executará o serviço.

Método: Objetivando-se criar um Perfil de usuário adequado à recomendação, esse experimento usou a abordagem de traços de personalidade (já apresentado na seção anterior) proposto por Nunes (Nunes 2009). Então, o Perfil/Reputação Psicológico do Usuário foi criado considerando o teste de personalidade NEO-IPIP (Johnson 2001) baseado em 300 questões.

Cerca de 100 pessoas foram convidadas a participar do experimento. Cada pessoa que participou do experimento foi instruída a responder o NEO-IPIP 3 vezes: (i) uma para o “candidato ideal”, as respostas refletiam como cada pessoa via o que considera características psicológicas de um presidente ideal; (ii) outra para o candidato “Nicolas Sarkozy”, isto é, cada pessoa respondeu o questionário informando como via e o que pensava sobre as características de tal candidato (Reputação); (iii) e outra para a candidata “Ségolène Royal”, isto é, cada pessoa respondeu o questionário informando como via e o que pensava sobre as características de tal candidato (Reputação).

Através das respostas, a modelagem dos aspectos psicológicos dos candidatos, isto é, o Perfil de usuário (Reputação), foi criado. A recomendação⁶ foi feita baseada nesses aspectos psicológicos.

Objetivando validar a precisão da recomendação gerada pelo Sistema de Recomendação, foi requisitado a cada pessoa que participou do experimento gentilmente confirmasse se o resultado da recomendação realmente havia representado o seu *voto real*.

Resultados: Apenas 10% das pessoas requisitadas responderam totalmente o questionário. O Sistema de Recomendação forneceu 2 tipos de recomendação. A primeira recomendação foi baseada nas 30 facetas do Big Five (para mais informação ver Nunes 2009) e, a segunda foi baseada simplesmente nas 5 dimensões do Big Five.

Os resultados da recomendação foram muito mais satisfatórios e representativos que o esperado. Os resultados são: (i) considerando a primeira recomendação, mais precisa (baseada nas facetas), a recomendação foi 100% precisa. Isto é, todas as recomendações geradas pelo Sistema de Recomendações combinaram com os votos efetivos dos participantes; (ii) considerando a segunda recomendação, menos precisa (baseado nas dimensões), a recomendação atingiu 80% de acerto. Isto é, houve 20% de casos onde a recomendação gerada pelo Sistema de Recomendação foi incompatível com o voto efetivo dos participantes.

O resultado demonstrou que usando uma complexidade maior na representação do Perfil Psicológico do usuário, no caso das facetas, o resultado obtido foi muito satisfatório.

O sucesso desse experimento referenciou o grande potencial no uso de questões psicológicas para auxiliar na tomada de decisão computacional através do uso de Sistemas de Recomendação. Ele foi o indicio necessário para apostar nessa nova e potencial tendência a ser usadas nas futuras recomendações.

4.6. Conclusões

Considerando que os Sistemas de Recomendação constituem-se em uma área de pesquisa que apresenta uma série de desafios e oportunidades. Já faz algum tempo que as pesquisas na área saíram da academia e tomaram forma no mercado, sendo inserido como solução em grandes sites de *e-commerce* como *Amazon.com*, *Submarino.com*, e atualmente o próprio site de recuperação de

⁶ O Sistema de Recomendação usou a técnica de recomendação baseada em outros contextos, considerando a personalidade. A abordagem usada na implementação foi algoritmo de busca linear usando a técnica do vizinho mais próximo.

informação Google, implementou uma solução de recomendação, onde o usuário uma vez logado poderá recomendar uma página.

Vimos neste capítulo como os Sistemas de Recomendação dotados de traços de personalidade podem ser utilizados para que se possa conhecer melhor os hábitos de consumo e interesses dos usuários, e como este tipo de conhecimento pode ser empregado para fidelizar clientes *web*.

Ao longo deste capítulo foi introduzida a área Computação Afetiva, incluindo uma breve descrição dos aspectos que envolvem a afetividade enfatizando a Personalidade. Seguindo-se por uma descrição e exemplificação de como, porque e quando a Computação Afetiva, principalmente a personalidade, potencializa a tomada de decisão humana. Apresentou-se, também, as abordagens de personalidade existentes exemplificando as codificáveis em computadores. Discutiu-se como os aspectos de personalidade influenciam na identidade do usuário e como isso afeta seu perfil. Na seqüência, foi descrito os critérios de armazenamento bem como as metodologias existentes hoje para extração de personalidade por computadores. Ainda discutiu-se as formas existentes de tomada de decisão computacional enfatizando os Sistemas de Recomendação, focando em sua aplicação, principalmente em e-commerce e TV digital. Questões relativas a técnicas e estratégias de recomendação foram apresentadas com o resumo de seus algoritmos.

Esse estudo tem mostrado que empresas tem buscando cada vez mais personalizar a relação humano-computador e esta tendência é motivada principalmente pelos interesses do *e-commerce e e-services*. Porém para evoluirmos na área da Computação Afetiva é preciso encontrar as melhores técnicas para capturar a personalidade e o perfil dos usuários. Existem linhas de pesquisa na área da psicologia que acreditam que é possível definir um indivíduo pela sua personalidade, a partir da análise de suas características.

Com o propósito de definir a personalidade surgem várias abordagens, tais como humanística, cognitiva, tipos, traço entre outras. Com base nessas abordagens são construídos modelos para capturar a personalidade. Os trabalhos estudados em sua maioria fizeram uso dos modelos baseados no BIG FIVE/FFM o que aparenta ser uma tendência, visto que, trabalha com a abordagem de traços. As pesquisas, que de modo geral utilizaram experimentos com um número pequeno (em média 60, entre homens e mulheres) de pessoas, demonstram que estes inventários são extremamente necessários à evolução da classificação das recomendações, porém, há estudos que comprovam que estas técnicas são formas invasivas e que desestimulam usuários a respondê-los, por seu extenso volume de questionamentos. Com isso, novos modelos têm sido sugeridos com a finalidade de melhorar a usabilidade do preenchimento destes formulários, associados às ferramentas probabilísticas capazes de otimizar a recomendação, ressaltando a necessidade de pesquisa com outras técnicas estatísticas que se adéquam ao contexto.

Dispositivos de aquisição de dados como teclado e mouse são soluções que não oneram os sistemas e nas pesquisas apresentadas, tem obtido resultados importantes na definição da personalidade e dos estados emocionais. Sensores de detecção de sinais fisiológicos, apesar de onerosos, possuem alta precisão na detecção destas atividades fisiológicas e melhoram bastante a classificação dos estados emocionais. Vimos, também a possibilidade de reconhecer e extrair traços de personalidade através de outra abordagem como diálogos e o uso de modelos estatísticos advindos da técnica de reconhecimento de padrões. Notoriamente, a importância de se reconhecer e responder traços de personalidade automaticamente possibilitaram a automatização nos processos de comunicação, derivando aplicabilidade variada em áreas da ciência, tecnologia e social.

O Sistema de Recomendação é uma tendência mundial quando estamos tratando de problemas que necessitam de apoio à decisão. Os Sistema de Recomendação, de uma forma geral,

tentam melhorar suas recomendações utilizando formulários de intervenção a usuários que buscam serviços e produtos.

Contudo, evidencia-se que um Sistema de Recomendação associado à Filtragem Colaborativa, onde as técnicas probabilísticas garantem uma maior acurácia à recomendação pela medida de similaridade dos usuários, tendo ainda Hardware e Software agregados aos sistemas com finalidades de enriquecer as definições dos modelos de usuário, modelos de itens e modelos de personalidades, são desejos da comunidade de Computação Afetiva na tentativa aproximar cada vez mais a um cenário real.

Alguns desafios emergem neste contexto, tais como: uso da computação afetiva com a utilização de agentes de software ou agentes de comunicação para uma maior interação com o usuário; criação de sistemas eficazes para recomendação a partir da aplicação de traços de personalidade adaptando a interface às reais necessidades psico-cognitivas do usuário, entre desafios que estão ainda surgindo dentro de uma área extremamente promissora no que tange a pesquisa.

Agradecimentos

Gostaríamos de agradecer a todos os alunos (IC, IT, TCC e Mestrado) que orientamos na área de Sistemas de Recomendação e contribuíram na elaboração do material de pesquisa usado na composição desse capítulo, principalmente ao aluno Luiz Trevisan que trabalhou o protótipo TV Plus, o aluno Jonas Santos Bezerra que está desenvolvendo o PersonalityML e a aluna Sandy Porto + Wanderson Costa que estão desenvolvendo um aplicativo para a extração automática de personalidade.

4.7. Referências Bibliográficas

- Adomavicius, Gediminas E Tuzhilin, Alexander. (2011) Context-Aware Recommender Systems. In: Ricci, Francesco Et. Al. Recommender Systems Handbook. Cap. 7, P. 217-253.
- Adomavicius, Gediminas E Tuzhilin, Alexander. (2005) Toward The Next Generation Of Recommender Systems: A Survey Of The State-Of-The-Art And Possible Extensions. In: Ieee Transactions On Knowledge And Data Engineering, P. 734-749, Los Alamitos.
- Allport, G. W.(1927). Concepts of trait and personality. Psychological Bulletin, (24):284–293. (Available at <http://psychclassics.yorku.ca/Allport/concepts.htm>).
- Andrade, L.O. S.; Nunes, M. A. S. N. (2011) Computação Afetiva: uma breve análise sobre os Sistemas de Recomendação Baseado em Conteúdo e Sistemas de Recomendação de Filtragem Colaborativa. Relatório Técnico- Notas de Mestrado. PROCC-Universidade Federal de Sergipe.
- Ansari, Asim; Essegari, Skander E Kohli, Rajeev. (2000) Internet Recommendation Systems. In: Journal Of Marketing Research, P. 363-375, Chicago.
- Araujo, Sandra R. C. E Carvalho, Victor T. (2011) Emuladores Para Tv Digital - Openmhp E Xletview. Disponível Em: <<http://Www.Tvdi.Inf.Br/Upload/Artigos/Artigo7.pdf>> Acesso Em: 10 Mar.
- Balabanovic, Marko E Shoham, Yoav Fab. (1997) Content-Based, Collaborative Recommendation. In: Communications Of The Acm, P. 66-72, Nova Iorque.
- Bezerra, J. S., Nunes. M. A. S. N., Oliveira. A. A. (2011) Desenvolvimento de metodologias de extração de perfil psicológico de usuário para aplicação em Sistemas de Recomendação objetivando personalização de produtos e serviços em e-commerce. Relatório Técnico de Pesquisa, Universidade Federal de Sergipe.
- Boyd, D. (2002). Faceted id/entity: Managing representation in a digital world. Master's thesis, Cambridge, MA.
- Burger, J.M. (2000). Personality. Wadsworth, fifth edition.
- Burke, Robin. (2002) Hybrid Recommender Systems: Survey And Experiments. In: User Modeling And User-Adapted Interaction, P. 331-370, Hingham.
- Carreira, R., Crato, J.M., Gonçalves, D. and Jorge, J. A. (2004). Evaluating adaptive user profiles for news classification. In IUI '04: Proceedings of the 9th international conference on Intelligent user interfaces, pages 206–212, New York, NY, USA. ACM Press.
- Cazella S. C., Nunes, M. A. S. N., Reategui, E. A. A. (2010) Ciência Da Opinião: Estado Da Arte Em Sistemas De Recomendação. In: Wagner Meira Jr. E André C. P. L. F. De Carvalho(Org.).. (Org.). Jai: Jornada De Atualização Em Informática Da Sbc. Rio De Janeiro: Editora Da Puc Rio, V. , P. 161-216.

- Cazella, S. C.; Correa, I. ; Barbosa, J. ; Reategui, E. (2009) Um Modelo Para Recomendação De Artigos Acadêmicos Baseado Em Filtragem Colaborativa Aplicado À Ambientes Móveis. *Revista Novas Tecnologias Na Educação*, V. 7, P. 12-22.
- Cazella, S. C. (2006) Aplicando A Relevância Da Opinião De Usuários Em Sistema De Recomendação Para Pesquisadores. Tese (Doutorado Em Ciência Da Computação) - Universidade Federal Do Rio Grande Do Sul, Porto Alegre, RS.
- Chanel, G. (2009) Emotion assessment for affective computing based on brain and peripheral signals. Thèse de doctorat : Univ. Genève.
- Damasio, Antonio R. (1994) *Descartes' Error: Emotion, Reason, And The Human Brain*. Quill, New York.
- Davis, F. D.; Bagozzi, R. P. and Warshaw, P. R.. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Manage. Sci.* 35, 8 982-1003.
- Donath, J. S. (2000). Being Real: Questions of Tele-Identity. In: Ken Goldberg, editor, *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*, chapter 16, pages 296–311. The MIT Press, first edition.
- Donath, J.S. (1999) Identity and Deception in the Virtual Community. In M. A. Smith and P. Kollock, editors, *Communities in Cyberspace*, chapter 2, pages 29–59. Routledge, London, first edition.
- Dourish, Paul. (2004) What We Talk About When We Talk About Context. In: *Personal And Ubiquitous Computing*. P. 19-30.
- Dunn, G., Wiersema, J., Ham, J., and Aroyo, L. (2009) Evaluating Interface Variants on Personality Acquisition for Recommender Systems. In *Proceedings of the 17th international Conference on User Modeling, Adaptation, and Personalization: Formerly UM and AH*. G. Houben, G. Mccalla, F. Pianesi, and M. Zancanaro, Eds. Lecture Notes In Computer Science, vol. 5535. Springer-Verlag, Berlin, Heidelberg, 259-270. (2009)
- Erikson, Erik H.(1980). *Identity and the Life Cycle*. Norton.
- Khan, I., Et Al. (2008) Measuring Personality from Keyboard and Mouse Use. *ACM International Conference Proceeding Series*, Vol. 369. Portugal.
- Giddens, A. (1991) *Modernity and Self-Identity. Self and Society in the Late Modern Age*. Stanford university Press, Stanford, California.
- Gill, A. J. ; Oberlander, J. (2003). Perception of e-mail personality at zero acquaintance: Extraversion takes care of itself; Neuroticism is a worry. *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 456–461). Hillsdale, NJ: LEA.
- Giordani, Alexandre. (2006) Wine Bunch: Uma Ferramenta Baseada Em Agrupamento Para Auxiliar Na Identificação De Contextos De Interesses De Consumidores Em *E-commerce*. Trabalho De Conclusão De Curso (Graduação Em Análise De Sistemas) - Universidade Do Vale Do Rio Dos Sinos, São Leopoldo, RS.
- Goffman, E. (1959). *The Presentation of Self in Everyday Life*. Anchor Book.
- Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, R. C., Gough, H. G., (2006) The international personality item pool and the future of public-domain personality measures. *Journal of Research in Personality* 40 (1), 84-96.
- Gosling, S. (2008) *Psiu, Dê Uma Espiadinha! Editora Campus*.
- Gonzalez, G., De La Rosa, J.L., And Montaner, M. (2007) Embedding Emotional Context Inrecommender Systems. In *The 20th International Florida Artificial Intelligence Research Society Conference-Flairs*, Key West, Florida.
- Heckmann, D. (2005). *Ubiquitous User Modeling*. Phd thesis, Technischen Fakultäten der Universität des Saarlandes, Saarbrücken-Germany.
- Heckmann, D. and Kruger, A. (2003). A user modeling markup language (UserML) for ubiquitous computing. In *8th International Conference on User Modeling, LNAI 2702*, page 393-397, Johnstown, PA, USA. Springer, Berlin Heidelberg.
- Herlocker, Jonathan L. (2000) *Understanding And Improving Automated Collaborative Filtering Systems*. Tese - Universidade De Minnesota, Minnesota.
- Hu,R and Pu,P. (2009a). A comparative user study on rating vs. personality quiz based preference elicitation methods. In *Proceedings of the 14th international conference on Intelligent user interfaces (IUI '09)*. ACM, New York, NY, USA, 367-372.
- Hu,R and Pu,P. (2009). Acceptance issues of personality-based recommender systems. In *Proceedings of the third ACM conference on Recommender systems (RecSys '09)*. ACM, New York, NY, USA, 221-224. DOI=10.1145/1639714.1639753 <http://doi.acm.org/10.1145/1639714.1639753>
- Hussain, M. S.; Calvo, R. A. (2009). A Framework for Multimodal Affect Recognition. Learning Systems Group, DECE, University of Sydney.
- Jensen, C.; Davis, J. and Farnham, S. (2002). Finding others online: reputation systems for social online spaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 447-454, New York, NY, USA. ACM.

- John, Oliver P. E. Srivastava, Sanjay. (1999) The Big Five Trait Taxonomy: History, Measurement, And Theoretical Perspectives. In: Handbook Of Personality: Theory And Research, P. 102–138. 1999. Nova Iorque.
- Johnson, J.A. (2000) Web-based personality assessment. In 71st Annual Meeting of the Eastern Psychological Association, Baltimore, USA. (Available at <http://www.personal.psu.edu/~j5j/vita.html>).
- John A. Johnson. (2001) Screening massively large data sets for nonresponsiveness in web-based personality inventories. Invited talk to the joint Bielefeld-Groningen Personality Research Group, University of Groningen, The Netherlands. (Available at <http://www.personal.psu.edu/faculty/j/5/j5j/papers/screening.html>).
- Johnson, J.A. (2005) Ascertaining the validity of individual protocols from web-based personality inventories. *Journal of research in personality*, 39(1):103–129.
- Josang, A., Ismail, R. and Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644.
- Khan, I. A.; Brinkman, W. P.; Fine, N.; Hierons, R. M. (2008) Measuring Personality from Keyboard and Mouse Use, In: European Conference in Cognitive Ergonomics, Madeira, Portugal.
- Kobsa, A. (2007). Generic user modeling systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter 4, pages 136–154. Springer Verlag.
- Konstan, J. A.; Miller, B. N., Maltz, D.; Herlocker, J. L.; Gordon, L. R. and Riedl, J. (1997). GroupLens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77-87.
- Lisetti, Christine L. (2002). Personality, affect and emotion taxonomy for socially intelligent agents. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 397–401. AAAI Press.
- Mairesse, F.; Walker, M..(2006) Automatic recognition of personality in conversation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, New York City.
- Mairesse, F.; Walker, M..(2008) A personality-based framework for utterance generation in dialogue applications. In *Proceedings of the AAAI Spring Symposium on Emotion, Personality and Social Behavior*.
- Mairesse, F., Walker, M.A., Mehl, M.R., Moore, R.K. (2007) Using Linguistic Cues for the Automatic Recognition of Personality in Conversation and Text. *Journal of Artificial Intelligence Research* 30, p. 457-500.
- Mahout. Apache Mahout. (2011) Disponível Em: <[Http://Mahout.Apache.Org](http://Mahout.Apache.Org)>. Acesso Em: 03 Mar.
- McDonald, David W. (2003). Recommending collaboration with social networks: a comparative evaluation. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–600, New York, NY, USA. ACM.
- Mead, G. H. (1934). *Mind, Self, and Society*, volume 1. University of Chicago, Chicago, Charles W. Morris edition.
- Nunes, M. A. S. N. ; Cerri, Stefano A. ; Blanc, N.(2008) . Improving Recommendations By Using Personality Traits In. In: *International Conference On Knowledge Management-In: International Conference On Knowledge Management-I.Know08, 2008, Graz-Austria. International Conference On Knowledge Management-I.Know08. V. 1. P. 92-100.*
- Nunes, M. A. S. N. (2009) Recommender Systems Based On Personality Traits: Could Human Psychological Aspects Influence The Computer Decision-Making Process?1. Ed. Berlin: Vdm Verlag Dr. Müller. V.1. 140p.
- Nunes, M. A. S. N. ; Aranha, C. N. (2009) .Tendências à Tomada de Decisão computacional . In: *W3C, 2009, São Paulo. W3C.*
- Nunes, Maria A. S. N. Et Al. (2010) Computação Afetiva E Sua Influência Na Personalização De Ambientes Educacionais: Gerando Equipes Compatíveis Para Uso Em Avas Na Ead. In: *Gláucio José Couri Machado. (Org.). Educação E Ciberespaço: Estudos, Propostas E Desafios. Aracaju: Virtus Editora, V. 1, P. 308-347.*
- Nunes, M. A. S. N. ; Bezerra, J. S. ; Oliveira, A. (2010b) Estendendo O Conhecimento afetivo da EmotionML. In: *IHC, 2010, Belo Horizonte. IHC2010. Porto Alegre : SBC.*
- Paiva, A. and Self, J.A. (1995). Tagus - a user and learner modelling workbench. *User Model. User-Adapt. Interact.*, 4(3):197–226.
- Perugini, Saverio; Gonçalves, Marcos André and Fox, Edward A. (2004). Recommender systems research: A connection-centric survey. *Journal of Intelligent Information Systems*, 23(2):107–143.
- Pianesi, F. ; Mana, N.; Cappelletti, A. ; Lepri, B.; Zancanaro, M..(2008) Multimodal recognition of personality traits in social interactions. in *Proceedings of the 10th International Conference on Multimodal Interfaces: Special Session on Social Signal Processing, 2008, pp. 53–60.*
- Picard, R. W. (1997) *Affective Computing*. Mit Press, Cambridge, Ma, Usa.
- Pina, E. da C.; Nunes, M.A.S.N. (2011) Os Rumos Da Pesquisa Científica Em Computação Afetiva. Relatório Técnico- Notas de Computação Afetiva. Mestrado-PROCC-Universidade Federal de Sergipe.

- Poo, D., Chng, B. and Goh, J.M. (2003). A hybrid approach for user profiling. In HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) Washington, DC, USA. IEEE Computer Society.
- Porto, S. M.; Costa, S. W.; Nunes, M. A. S. N; Matos, L. N. (2011) Desenvolvimento de Metodologias de Extração de Perfil Psicológico de Usuário para Aplicação em Sistemas de Recomendação Objetivando Personalização de Produtos e Serviços em E-Commerce. Relatório Técnico de Pesquisa. Universidade Federal de Sergipe.
- Rabelo, R. A. C.; Nunes, M.A.S.N. (2011) Um Estudo Sobre Modelos E Frameworks De Reconhecimento De Personalidade Em Diálogos. Relatório Técnico- Notas de Computação Afetiva. Mestrado-PROCC-Universidade Federal de Sergipe.
- Recio-Garcia, J. A.; Jimenez-Diaz, G.; Sanchez-Ruiz, A. A. and Diaz-Agudo. B. (2009). Personality aware recommendations to groups. In Proceedings of the third ACM conference on Recommender systems (RecSys '09). ACM, New York, NY, USA, 325-328.
- Reeves, B. Nass, C. (1996) The Media Equation: How People Treat Computers, Television, And New Media Like Real People And Places. Cambridge University Press, New York, Ny, Usa.
- Rein, G. L. (2005). Reputation information systems: A reference model. In HICSS'05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1, page 26, Washington,DC, USA. IEEE Computer Society.
- Resnick, P.; Zeckhauser, R.; Swanson, J. and Lockwood, K.. (2006). The value of reputation on ebay: A controlled experiment. *Experimental Economics*,9(2):79-101.
- Resnick, P.; Kuwabara, K.; Zeckhauser, R. and Friedman, E. (2000). Reputation systems. *Commun. ACM*, 43(12):45-48.
- Resnick, Paul; Varian, Hal R. (1997) Recommender Systems. In: *Communications Of The Acm*. P. 56-58, Nova Iorque.
- Riedl, J. Et Al. (1997) GroupLens: Applying Collaborative Filtering To Usenet News. *Communications Of The Acm*, New York, V.40, N.3, Pp. 77-87, Mar.
- Rousseau, B., Browne, P. , Malone, P. Foster, P. and Mendis, V. (2004). Personalised resource discovery searching over multiple repository types: Using user and information provider profiling. In *ICEIS (5)*, pages 35–43.
- Salton, G., McGILL J. M. (1983) Introduction to Modern Information Retrieval. McGraw Hill, New York, USA.
- Sandvig, J. J., Mobasher, Bamshad E Burke, Robin. (2007) Robustness Of Collaborative Recommendation Based On Association Rule Mining. In: *Acm Conference On Recommender Systems*. P. 105-112.
- Schafer, J. B.; Konstan, J. and Riedl, J. (1999). Recommender systems in e-commerce. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 158-166, New York, NY, USA. ACM.
- Schafer, J. B.; Konstan, J. and Riedl, J. (2001). E-commerce recommendation applications. *Data Mining Knowledge Discovering*, 5(1-2):115-153.
- Schultz, D. (1990) Theories of Personality. Brooks/Cole, forth edition.
- Simon, H.A. (1983) Reason In Human Affairs. Stanford University Press, California.
- Thagard, Paul. (2006) Hot Thought: Mechanisms And Applications Of Emotional Cognition. A Bradford Book- Mit Press, Cambridge, Ma, Usa.
- Tkalčič, Marko Et. Al. (2009) Personality Based User Similarity Measure For A Collaborative Recommender System. In: *The 5th Workshop On Emotion In Human-Computer Interaction*, P. 279-311, Cambridge.
- Tkalčič, M.; Burnik, U.; Košir, A. (2010) Using affective parameters in a content-based recommender system for images. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, Volume 20, Number 4, pages: 279-311.
- Trappl, Robert; Payr, Sabine And Petta, Paolo Editors. (2003) Emotions In Humans And Artifacts. Mit Press, Cambridge, Ma, Usa.
- Trevisan, Luiz. (2011) Tv Plus: Um Modelo De Sistema De Recomendação De Programas Para Tv Digital. Trabalho De Conclusão De Curso (Graduação Em Sistemas De Informação) - Universidade Do Vale Do Rio Dos Sinos, São Leopoldo, RS.
- Salton, G., McGill J. M. (1983) Introduction To Modern Information Retrieval. McGraw Hill, New York.
- Soldz, Stephen E Vaillant, George E. (1998) The Big Five Personality Traits And The Life Course: A 45 Years Longitudinal Study. In: *Journal Of Research In Personality*, P. 208-232.
- W3C 2010a. Emotion Markup Language (EmotionML) 1.0 W3C - Working Draft 29 October 2009. Disponível em: <http://www.w3.org/TR/2009/WD-emotionml-20091029/>. Acesso em 08/06/2010.
- W3C 2010b. SOAP - W3C Recommendation. <http://www.w3.org/TR/soap/> . Acesso em 01/07/2010.
- W3C 2010c. EMMA: Extensible MultiModal Annotation markup language version 1.0. <http://www.w3.org/TR/emma/> . Acesso em 01/07/2010.
- W3C 2010d. XML Specification. WorldWideWeb Consortium. Disponível em <http://www.w3c.org>. Acesso em 01/07/2010.

Capítulo

5

Por dentro das redes complexas - detectando grupos e prevendo ligações

Ana Paula Appel¹, Estevam Rafael Hruschka Junior²

¹Departamento de Engenharias e Computação – DECOM
Universidade Federal do Espírito Santo - UFES
Centro Universitário Norte do Espírito Santo - CEUNES
{anaappel@ceunes.ufes.br}

²Departamento de Computação - DC
Universidade Federal de São Carlos - UFSCar
{estevam@dc.ufscar.br}

Abstract

The growing volume of data modeled as complex networks, eg the World Wide Web, social networks like Orkut, Facebook, has raised a new area of research - mining complex networks. In this new multidisciplinary area some tasks can be highlighted: the extraction of statistical properties, community detection and link prediction. This short course aims to introduce not only the basics of mining complex networks, but also the techniques of community detection and link prediction. The mining of complex networks have not only been the focus of a large number of researchers but also big companies like Microsoft, Google, Yahoo and Facebook.

Resumo

O crescimento do volume de dados modelados como redes complexas, por exemplo a World Wide Web, redes sociais como Orkut, Facebook, fez surgir uma nova área de pesquisa - a mineração de redes complexas. Nesta nova área multidisciplinar destacam-se algumas tarefas: a extração de propriedades estatísticas, a detecção de comunidades, a predição de ligações arestas. Este minicurso tem como objetivo introduzir não só os conceitos básicos da mineração de redes complexas, mas também as técnicas de detecção

de comunidade e predição de ligação. A área de mineração de redes complexas têm sido o foco não só de um grande número de pesquisadores mas também de grandes empresas como Microsoft, Google, Facebook e Yahoo.

5.1. Introdução

De uma maneira simplista, uma rede é uma coleção de pontos conectados por pares de linhas. Dependendo da área de utilização os pontos recebem nomes específicos, como nós ou vértices e as linhas são referenciadas como arestas. Muitos objetos de interesses têm sido mapeados como redes, por exemplo, as redes sociais (Facebook, Flickr, Orkut, etc), redes biológicas, redes de computadores, ontologias, redes acadêmicas (DBLP, ARXIV) e a própria WWW - World Wide Web.

Tais redes, chamadas redes complexas são, normalmente, modelada como um grafo, ou seja, a rede complexa é representada por meio de um objeto matemático cujos nós, também chamados vértices, modelam elementos (que podem ser páginas web, pessoas, computadores) e as arestas modelam relacionamentos entre os nós. Este tipo de representação é conveniente em muitas situações, já que ela abstrai o problema para uma representação onde apenas a conexão entre os objetos é levada em consideração eliminando muitas vezes a complexidade e informações desnecessárias. Informações adicionais podem ser inseridas, como nomes e pesos aos nós e arestas [Newman, 2010].

Nos últimos anos as redes complexas têm sido o foco de estudo de muitos cientistas, os quais vêm desenvolvendo uma ampla coleção de ferramentas para modelar, analisar e entender as redes complexas. Estas ferramentas são em sua maioria matemáticas, estatísticas e computacionais e muitas vezes partem de uma simples representação da rede com o conjunto de nós e arestas e depois de alguns cálculos produzem informações interessantes sobre ela, como por exemplo o caminho médio entre pares de nós, quantidade de triângulos, etc [Faloutsos et al., 1999, Leskovec et al., 2007, Tsourakakis, 2008].

A ciência moderna das redes complexas tem trazido um avanço significativo no entendimento de sistemas complexos. Uma das mais relevantes características das redes que representam sistemas reais é a estrutura de comunidade, ou “clustering”, isto é, a organização de vértices em grupos, com muitas arestas unindo vértices do mesmo grupo e comparativamente poucas arestas unindo vértices de grupos diferentes. Tais grupos, ou comunidades, podem ser consideradas partes independentes das redes, desempenhando um papel semelhante aos tecidos ou órgãos do corpo humano. Detectar comunidades é de grande importância em áreas como a sociologia, biologia e ciência da computação, disciplinas nas quais frequentemente os sistemas são representados como redes complexas. Contudo, encontrar comunidades em redes complexas é um problema de difícil solução e ainda não possui uma resposta satisfatória, apesar do grande esforço que tem sido feito pela comunidade científica nos últimos anos [Fortunato, 2010].

Outro problema bastante comum na área de redes complexas é a predição de ligação. A predição de ligação tem a sua fonte de inspiração as redes sociais, na qual quer se prever com uma acurácia cada vez maior quem são os seus possíveis novos amigos de cada um dos membros das redes sociais [Liu and Wong, 2008]. A tarefa de predição de ligações (*Link Prediction*) tem por objetivo prever quais arestas irão surgir em uma rede complexa em um futuro próximo [Liben-Nowell and Kleinberg, 2003]. De modo mais

formal, a predição de ligações pode ser definida como, dado um “*snapshot*” de uma rede complexa em um tempo t , quer se prever com uma certa acurácia as arestas que irão surgir na rede complexa no tempo futuro $t + 1$. Uma das dificuldades da predição de ligações é que as redes complexas tendem a ser esparsas.

Com o intuito de guiar o leitor se tentará uma exposição minuciosa do tema neste documento, a partir da definição dos principais elementos do problema, para a apresentação da maioria dos métodos desenvolvidos, com um foco especial em técnicas mais recentes, a partir da discussão de questões cruciais como a importância do métodos de agrupamento e de predição de ligação e as principais diferenças entre os diferentes métodos de cada técnica. Além disso, sempre que possível será dada uma descrição de aplicações para as redes complexas reais.

5.2. Conceitos

As redes complexas, ou simplesmente redes, são um conjunto de elementos discretos que são representados por vértices e arestas, sendo que as arestas são um conjunto de conexões entre os vértices. Os elementos e suas conexões podem representar, por exemplo, pessoas e ligações de amizade, computadores e linhas de comunicação [Faloutsos et al., 1999], componentes químicos e reações [Jeong et al., 2000], artigos e citações [Redner, 1998], entre outros.

As redes complexas podem ser facilmente modeladas como um grafo. Os grafos são capazes de abstrair os detalhes do problema ao descreverem características topológicas importantes com uma clareza que seria praticamente impossível se todos os detalhes fossem mantidos. Essa foi uma das razões por que a teoria dos grafos se espalhou, especialmente nos últimos anos, e tem sido utilizada por engenheiros, cientistas da computação e em especial por sociólogos.

Nesta seção serão apresentados os conceitos como os da teoria dos grafos, álgebra linear e outros que se fazem necessários para a o entendimento das tarefas de detecção de comunidades e predição de ligação.

5.2.1. Teoria dos Grafos

Um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ é definido como um conjunto de nós \mathcal{V} e um conjunto de arestas \mathcal{E} , sendo que $|\mathcal{V}| = N$ denota o número de nós e $|\mathcal{E}| = M$ denota o número de arestas, sendo $e_k \in \mathcal{E}$ e $e_k = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}\}$. Os termos nó ou vértice são considerados sinônimos. Neste trabalho será usado o termo nó para referenciar os elementos do conjunto de vértices \mathcal{V} e similarmente o termo aresta para referenciar os elementos do conjunto de arestas \mathcal{E} , que também é referenciado na literatura por meio dos seguintes sinônimos: *links*, *hops*, ligações ou conexões.

Uma maneira conveniente de representar um grafo \mathcal{G} em um computador é usar uma matriz de adjacência, que é uma matriz \mathbf{A} quadrada $N \times N$, sendo $N = |\mathcal{V}|$, em que $\mathbf{A}_{i,j} = 1$ se $(v_i, v_j) \in \mathcal{E}$ e 0 caso o contrário.

Seja S uma comunidade na qual n_S é o número de nós em S e $n_S = |S|$; m_S o número de aresta em S e $m_S = |(u, v) : u \in S, v \in S|$ e c_S o número de arestas na fronteira de S , sendo $c_S = |(u, v) : u \in S, v \notin S|$

Tabela 5.1. Símbolos utilizados neste trabalho.

Símbolo	Descrição
\mathbf{A}	Matriz de Adjacência
\mathcal{G}	grafo
\mathcal{E}	arestas
\mathcal{V}	nós
λ	autovalor do grafo
v_i	nó de um grafo
e_k	aresta de um grafo
Δ	triângulo
$d(v_i)$	grau do nó v_i
$\Gamma(v_i)$	grau do nó v_i
$d_{out}(v_i)$	grau de saída do nó v_i
$d_{in}(v_i)$	grau de entrada do nó v_i
d_{max}	maior grau do grafo
$P(v, u)$	caminho do nó v ao u
$C(v_i)$	coeficiente de clusterização do nó v_i
$C(\mathcal{G})$	coeficiente de clusterização do grafo
$score(u; w)$	valor entre dois nós
κ_t	clique de tamanho t
k	número de nós em uma comunidade
S	comunidade
cS	número de arestas na fronteira de S
N	número de nós
nS	número de nós em S
M	número de arestas
mS	número de arestas em S

A Tabela 5.1 apresenta os principais símbolos utilizados e a seguir apresentam-se alguns conceitos básicos, extraídos de [Nicoletti, 2006, Bondy and Murty, 1979, Diestel, 2005], que serão usados neste minicurso:

- **Grafos Direcionados e Não Direcionados:** um grafo é *não direcionado* se $\{(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}\}$, isto é, as arestas são pares de nós sem ordem. Se um par de nós é ordenado, isto é, arestas tem direção, então o grafo é *direcionado*, também chamado de *dígrafo*.
- **Grado do Nó:** o nó v_i tem grau $d(v_i)$ também representado por $\Gamma(v_i)$ se ele tem $|\mathcal{N}(v_i)|$ nós incidentes. Para grafos direcionados, o grau de um nó pode ser dividido em “grau de saída”, $d_{out}(v_i)$ que é o número de arestas entram no nó v_i e “grau de entrada”, $d_{in}(v_i)$ que é o número de arestas que saem para o nó v_i .
- **Triângulo:** em um grafo não direcionado um triângulo (Δ), também conhecido como fechamento transitivo, é uma tripla de nós conexos (u, v, w) , tal que, $(u, v), (v, w), (w, u) \in \mathcal{E}$

- **Caminho:** é uma sequência de nós conectados entre si, $P(v_1, v_n) = (v_1, v_2, v_3, \dots, v_n)$, tal que, entre cada par de nó existe uma aresta $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n) \in \mathcal{E}$. Um caminho é **simples** se nenhum nó se repete. Dois caminhos são **independentes** se somente o primeiro e o último nó são comuns à eles.
- **Comprimento de um caminho:** é o número de arestas que o caminho contém. O **menor caminho** entre dois nós $P(v_i, v_j)$ é o caminho de menor número de arestas que ligam os dois nós.
- **Subgrafo:** um subgrafo $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ de um grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ é um subconjunto de arestas e todos os nós tal que $\mathcal{E}_s \subseteq \mathcal{E} \Rightarrow \mathcal{V}_s = \{v_i, v_j | (v_i, v_j) \in \mathcal{E}_s\}$.
- **Grafo Conexo:** é um grafo que possui pelo menos um caminho entre todos os pares de nós.
- **Componente Conexa:** é o maior subgrafo, na qual existe um caminho entre qualquer par de vértice.
- **Clique (κ_t):** é um subgrafo completo que possui um subconjunto de nós $\mathcal{V}_s \subseteq \mathcal{V}$ e arestas conectando todos os pares de nós em \mathcal{V}_s . O tamanho t do clique é definido pelo número de nós, $|\mathcal{V}_s| = t$. Um triângulo é um clique de tamanho 3 - κ_3 .
- **Diâmetro:** o diâmetro \mathcal{D} de um grafo \mathcal{G} é o maior caminho dentre todos os menores caminhos existentes entre todos os pares de nós do grafo \mathcal{G} .

5.3. Mineração de Grafos

Diversos domínios de aplicações têm seus dados modelados como redes complexas, por exemplo, a Internet, a World Wide Web (WWW), as redes sociais, de colaboração, biológicas, entre outras. Os pesquisadores nos últimos anos têm identificado classes de propriedades que podem ser encontradas em muitas das redes reais de vários domínios, sendo que muitas dessas propriedades são distribuições que seguem leis de potência, como a distribuição do grau dos nós, número de triângulos e os autovalores da matriz de adjacência da rede complexa.

Grafos que modelam redes que representam sistemas reais não são regulares como os reticulados, na qual todos os vértices possuem o mesmo grau, nem randômicas, na qual são redes em que cada aresta possui uma probabilidade existir igual para todos os pares de vértices possíveis fazendo com que a distribuição de arestas entre os vértices seja altamente homogênea. A Figura 5.1 apresenta duas redes reais com distribuições diferentes. A primeira é uma randômica que representa algumas conexões entre as rodovias do EUA (à direita da Figura 5.1) e a segunda uma rede que segue uma distribuição de potência que representa as conexões entre os aeroportos dos EUA (à esquerda da Figura 5.1).

Como visto na Figura 5.1, as redes reais que seguem uma lei de potência exibem uma grandes inomogeneidade, a distribuição do grau dos nós é ampla, com uma cauda que na maior parte das vezes segue uma lei de potência: por isso, muitos vértices de baixo grau coexistem com alguns vértices de alto grau. Uma lei de potência, que é uma

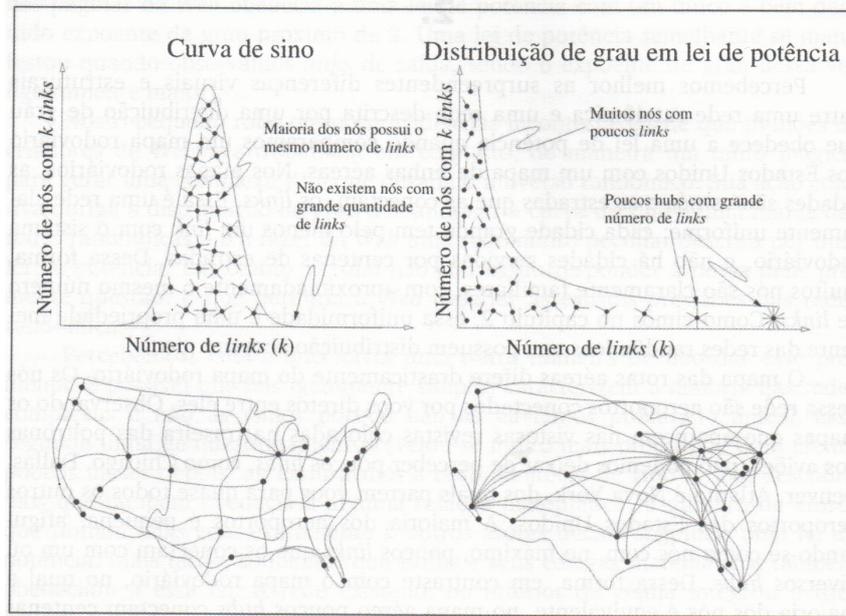


Figura 5.1. Comparação entre a distribuição dos graus dos nós das redes reais e randômicas [Barabasi, 2002]

distribuição na forma, $p(x) = a * x^{-\gamma}$, na qual $p(x)$ é a probabilidade de x ocorrer, sendo a uma constante de proporcionalidade e γ o expoente da lei de potência [Newman, 2005, Clauset et al., 2009].

A distribuição do grau dos nós de uma rede é uma lei de potência se o número de nós N_ϕ que possui um grau ϕ é dado por $N_\phi \propto \phi^{-y}$ ($y > 1$), sendo $\mathcal{V}_\phi = \{v_i \in \mathcal{V} | d(v_i) = \phi\}$, $|\mathcal{V}_\phi| = N_\phi$ e y é chamado de expoente da distribuição do grau. A grande maioria das redes reais apresentam uma distribuição do grau dos nós que segue uma lei de potência.

Além disso, a distribuição das arestas não é só globalmente, mas também localmente heterogênea, com altas concentrações de bordas dentro de grupos especiais de vértices, e baixas concentrações de entre estes grupos. Esta característica das redes reais é chamado de estrutura da comunidade [Girvan and Newman, 2002], ou clustering. Comunidades, também chamadas de clusters ou módulos, são grupos de vértices com alta probabilidade de compartilharem propriedades ou desempenharem um papel semelhante dentro da rede.

A sociedade oferece uma grande variedade de possíveis organizações em grupos: as famílias, os colegas de trabalhos e os círculos de amizades. A difusão da Internet também levou à criação de grupos virtuais, que vivem na Web, como as comunidades online (Facebook, Orkut, Google+, MySpace, etc). De fato, as comunidades sociais têm sido estudadas por um longo tempo [Coleman, 1988, Freeman, 2004]. Comunidades também ocorrer em muitas redes que representam sistemas da biologia, ciência da computação, engenharia, economia, política, etc. Em redes de interação proteína-proteína, as comunidades são como grupos de proteínas que têm a mesma função específica dentro da célula [Newman, 2006], a rede da World Wide Web pode corresponder a grupos de páginas com os mesmos temas ou tópicos relacionados [Flake et al., 2002], em redes me-

tabólicas podem estar relacionadas aos módulos funcionais, tais como ciclos e caminhos [Palla et al., 2005], e assim por diante.

Comunidades têm aplicações concretas, por exemplo, agrupar clientes Web que têm interesses semelhantes e são geograficamente próximos uns dos outros pode melhorar o desempenho dos serviços prestados à World Wide Web, em que cada grupo de clientes pode ser servido por um servidor espelho dedicado [Krishnamurthy and Wang, 2000]. Identificar grupos de clientes com interesses semelhantes na rede de relações de compra entre clientes e produtos de varejistas online (como a www.amazon.com) permite a criação de sistemas de recomendação eficientes [Reddy et al., 2002], que melhor orientam os clientes por meio de uma lista de itens do varejista e aumentam as oportunidades de negócio entre outras aplicações.

A detecção de comunidade é importante não só pela grande quantidade de aplicações, mas por outras razões também. Identificar os grupos e suas fronteiras permite uma classificação dos vértices de acordo com sua posição estrutural nos grupos. Assim, vértices com uma posição central em seus clusters, compartilhando um grande número de arestas com outros membros do grupo, podem ter uma importante função de controle e estabilidade dentro do grupo; vértices encontrados nas fronteiras entre os grupos têm um papel importante de mediação e conduzem as relações e intercâmbios entre as diferentes comunidades, na qual os sinais são transmitidos em todo o grafo seguindo caminhos de comprimento mínimo.

Na literatura os termos *particionamento de grafo* e *detecção de comunidades* são utilizados para referenciar a divisão dos vértices em grupos. Contudo, os termos podem ser diferenciados pela necessidade ou não de definir o número de grupos e objetos dentro do grupo.

5.3.1. Particionamento de Grafos

O particionamento de grafos é um problema clássico em ciência da computação, estudado desde a década de 60. O problema é a divisão dos vértices de uma rede em grupos não sobrepostos tal que o número de arestas entre os grupos seja minimizados. Neste caso o número de grupos e de vértices em cada grupo é fixo.

Um exemplo clássico é a alocação de tarefas em um processador para que a comunicação entre eles seja minimizada permitindo a alta performance em cálculos. Isto pode ser acoplado pela divisão de um cluster de computadores em grupos com aproximadamente o mesmo número de processadores, tal que o número de conexões físicas entre os processadores de diferentes grupos seja mínima.

Apesar de não ser o foco deste minicurso, o problema de particionamento de grafos é utilizado por alguns algoritmos de detecção de comunidades, assim será feita uma breve explanação sobre o assunto.

O problema mais simples de particionamento de grafos é a divisão de uma rede em duas, chamada *bisseção do grafo*. A maioria dos algoritmos de particionamento de grafos são de fato algoritmos de *bisseção*, já que após dividir uma rede em duas, esta pode ter cada uma das duas novas partes divididas em outras duas.

De modo mais formal, o problema de particionar um grafo consiste em dividir o

conjunto de vértices em k grupos de tamanho pré-definido, tal que, o número de arestas entre cada um dos grupos seja mínimo. O número de arestas removidas é chamado *tamanho de corte*. Especificar o número de grupos em que uma rede será particionada é necessário, pois se este número fosse deixado livre a resposta trivial seria uma única partição com todos os nós. A Figure 5.2 ilustra o particionamento de um grafo em dois grupos ($k = 2$) cada um com 7 nós.

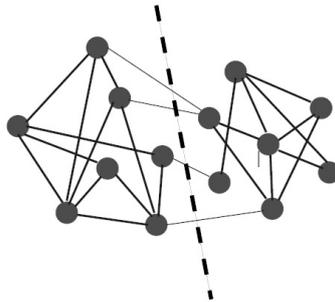


Figura 5.2. Um grafo sendo particionado em dois grupos iguais, cada um com 7 nós.

Apesar de ter uma definição simples, o problema em si não é tão simples de ser resolvido, já que o número de possibilidades de uma rede com N vértices ser dividida em dois grupos N_1 e N_2 é $N!/(N_1!N_2!)$.

O algoritmo mais tradicional para o particionamento de redes é o METIS, que permite o particionamento do grafo em k grupos. O algoritmo METIS funciona da seguinte maneira: dado um grafo \mathcal{G} , este é reduzido para um grafo com o agrupamento dos vértices adjacentes em um mesmo nó. A bisseção deste grafo muito menor é computada e então o particionamento é projetado no grafo original por meio de refinamentos sucessivos da partição [Karypis and Kumar, 1998]. A Figura 5.3 ilustra o funcionamento do algoritmo METIS em uma rede. Note que apesar de na ilustração apresentar apenas o particionamento da rede em dois grupos, o METIS permite que o grafo seja particionado em k grupos, sendo k definido pelo usuário.

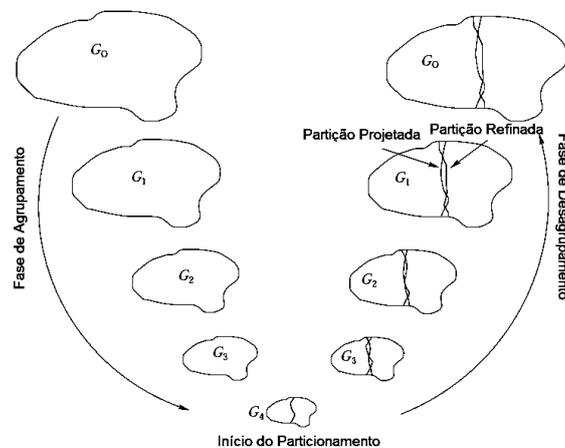


Figura 5.3. Exemplo de funcionamento do algoritmo METIS [Karypis and Kumar, 1998].

Outro método muito utilizado para o particionamento de grafos é o *Spectral Bi-*

section [Alon, 1998]. Este método é baseado no cálculo do segundo menor autovetor da matriz laplaciana da rede. Na teoria dos grafos, os autovalores λ_i de um grafo \mathcal{G} são definidos como sendo os autovalores da matriz de adjacência A do grafo \mathcal{G} ou a matriz laplaciana B . A matriz laplaciana é definida como B tem-se que $B_{i,j} = d(v_i)$ se $i = j$, $B_{i,j} = -1$ se $i \neq j \wedge (v_i, v_j) \in \mathcal{E}$ e 0 caso o contrário. Usando o segundo menor autovetor a matriz é reordenada e os grupos são identificados. Um exemplo deste método é apresentado na Figura 5.4. Primeiro a matriz de adjacência de um grafo \mathcal{G} , os pontos pretos indicam os uns na matriz e os espaço vazio os zeros, ao seu lado a matriz é reordenada pelo segundo menor autovetor da matriz laplaciana do grafo.

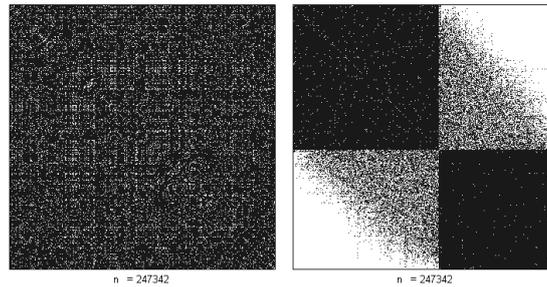


Figura 5.4. Matriz de adjacência de um grafo \mathcal{G} e a matriz reordenada pelo segundo menor autovetor da matriz laplaciana do grafo \mathcal{G}

5.3.2. Detecção de Comunidades

O problema da detecção de comunidades se difere do particionamento de grafos, pois nela o número de grupos e o número de elementos dentro dos grupos não são especificados pelo usuário. Ao contrário, este número é determinado pela própria rede, isto é, o objetivo da detecção de comunidades é encontrar grupos formados naturalmente pela estrutura topológica da rede. Além disso, o tamanho de cada grupo existente na rede pode variar abruptamente. A Figura 5.5 apresenta a estrutura tradicional de comunidades em redes complexas. Contudo, o problema de detecção de comunidade é muito menos definido do que o particionamento de grafos. A começar pela definição mais aceita que diz: "Um grupo é um conjunto de vértices que tende a ser mais conectado entre si e menos conectado com outros grupos". O que exatamente significa "mais" e "menos"? Para responder esta questão uma grande variedade de trabalhos têm sido propostos nos últimos anos.

Assim, a medida de "mais" aresta dentro da comunidade e "menos" aresta fora da comunidade é formalizada com a "densidade intra-cluster" ($\delta_{int}(S)$) e "densidade inter-cluster" ($\delta_{ext}(S)$). A densidade intra-cluster é o número de arestas que ligam dois vértices pertencentes a mesma comunidade dividido pelo número possível de arestas que possam existir dentro comunidade, $\delta_{int}(S) = \frac{\text{arestas internas de } S}{nS(nS-1)/2}$. A densidade inter-cluster é similarmente definida, sendo o número de arestas que conectam vértices da comunidade com o restante da rede dividido pelo número máximo de arestas inter-cluster que possa existir, $\delta_{ext}(S) = \frac{\text{arestas inter-cluster } S}{nS(n-nS)}$.

Para S ser uma comunidade é esperado que $\delta_{int}(S)$ ser um número maior que a densidade média da rede ($\delta(\mathcal{G})$), que é dada pelo número de arestas da rede dividido pelo

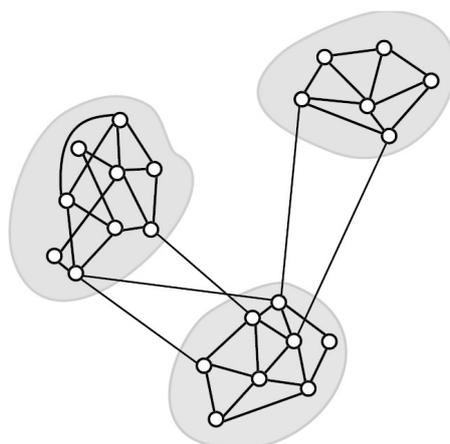


Figura 5.5. Grafo apresentando 3 comunidades do modo tradicional.

número máximo possível de arestas que possam existir na rede. Por outro lado, $\delta_{ext}(S)$ tem que ser muito menor que $\delta(\mathcal{G})$. A busca pela melhor combinação de um $\delta_{int}(S)$ maior e um $\delta_{ext}(S)$ menor está implícito ou explícito nos algoritmos de detecção de comunidades. Uma maneira simples de fazer isto é maximizar a soma das diferenças $\delta_{int}(S) - \delta_{ext}(S)$ sobre todos os grupos de uma partição [Mancoridis et al., 1998].

Uma propriedade indispensável de uma comunidade é a “*connectedness*”. Espera-se que para S ser uma comunidade deve haver um caminho entre cada par de vértice da comunidade percorrendo apenas vértices de S . Nesta definição um rede com diversas componentes conexas teria cada uma de suas componentes analisadas independentemente.

Existem muitas outras medidas baseadas em densidade que têm sido usadas para particionar um gráfico em um conjunto de comunidades [Brandes and Erlebach, 2005, Schaeffer, 2007, von Luxburg, 2007]. Uma que merece menção especial é a modularidade [Newman, 2006, Newman and Girvan, 2003]. Para uma determinada partição de uma rede em um conjunto de comunidades, a modularidade mede o número de arestas dentro da comunidade em relação a um modelo (*null model*), que é geralmente considerado como sendo um grafo randômico com o mesmo grau de distribuição. Assim, a modularidade foi originalmente introduzida e normalmente usada para medir a força ou a qualidade de uma partição específica de uma rede.

Em mais detalhes, considere uma rede composta por N nós ou vértices conectados por M arestas e seja A_{ij} um elemento da matriz de adjacência da rede, o que dá o número de arestas entre os vértices i e j . E suponha que é dada uma divisão candidata dos vértices em um certo número de grupos. A modularidade desta divisão é definida como a fração das arestas que se enquadram dentro dos grupos menos a dada fração esperada, caso as arestas sejam distribuídas aleatoriamente. Na versão mais comum do conceito, a randomização das arestas é feita a fim de preservar o grau de cada vértice. Neste caso, o número esperado de arestas compreendida entre dois vértices i e j seguindo a randomização é $k_i * k_j / 2M$ onde k_i é o grau do vértice i , e, portanto, o número real menos esperado de arestas entre os mesmos dois vértices é $A_{i,j} - k_i * k_j / 2M$. Assim, Q apresentado na Equação 1 é a soma sobre todos os pares (i, j) que estão na mesma comunidade.

$$Q = \frac{1}{2M} \sum_{ij} [A_{ij} - \frac{k_i * k_j}{2m}] (C_i, C_j) \quad (1)$$

Outra medida utilizada na detecção de comunidades é o *betwenness* introduzido por Girvan e Newman em [Girvan and Newman, 2002]. O *betwenness* é a medida que calcula a importância de uma aresta baseado no número de caminhos mínimos entre pares de nós que passam por ela, ou seja, quanto mais caminhos mínimos passarem por uma aresta para conectar dois nós maior será a importância da aresta. No algoritmo proposto por Girvan e Newman, a detecção de comunidades é feita com sucessivas remoções de arestas que conduzem ao isolamento da comunidade, as arestas a serem retiradas são as que possuem maior *betwenness*. Um exemplo desse processo é apresentado na Figura 5.6

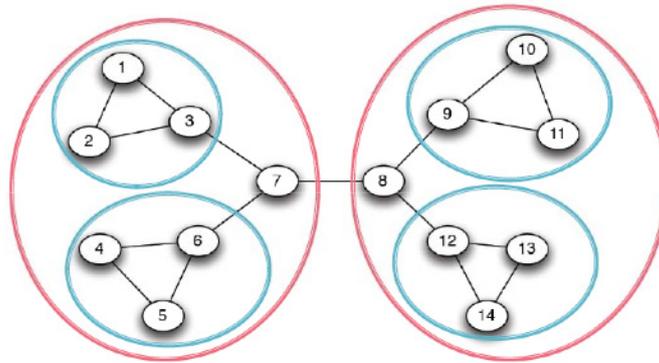


Figura 5.6. Cada um dos círculos representam a definição de uma comunidade. A aresta que conecta os nós 7 e 8 é a que possui a maior *betwenness*, 49, sendo a primeira aresta a ser retirada e dividindo a rede em duas comunidades. O processo de retirada de aresta continua até se encontrar as comunidades desejadas.

Além do *betwenness* e da modularização outra medida utilizada é a condutância. A condutância é uma das medidas mais simples já que leva em consideração o número de arestas dentro e fora da comunidade [Kannan et al., 2000, Leskovec et al., 2008]. Formalmente a condutância $\varphi(S)$ para um conjunto de nós S é $\varphi(S) = cS / \min(\text{Vol}(S), \text{Vol}(V \setminus S))$, sendo cS o número de arestas na fronteira, $cS = |(u, v) : u \in S, v \in S|$ e $\text{Vol}(S) = \sum_{u \in S} d(u)$ sendo $d(u)$ é o grau do nó u . Assim, a condutância consegue capturar a noção de comunidades de nós que tenham uma melhor conectividade interna do que externa. A Figura 5.7 mostra as comunidades A, B, C, D e E. O grupo de nós B é mais propício a ser comunidade do que A, pois, $\varphi(A) = 2 > \varphi(B) = 1$.

Uma técnica que se baseia na condutância é o *network community profile* (NCP) que caracteriza a qualidade das comunidades encontradas na rede baseando-se em seu tamanho. Para cada k entre 1 e metade do número de nós da rede é definido $\Phi(k) = \min_{|S|=k} f(S)$. O que significa que, para cada possível tamanho de comunidade k , $f(k)$ mede o valor da melhor comunidade de tamanho k e o NCP mede $\Phi(k)$ em função de k . A Figura 5.8 apresenta a melhor condutância para comunidade com $k = 4$ que é B, já que $\Phi(4) = \frac{1}{11}$. Similarmente, D e D+E apresentam a melhor condutância para 3 e 6 nós.

Para grandes redes reais foi observado também que o mínimo global é alcançado

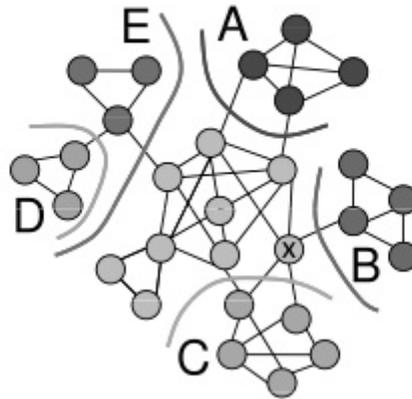


Figura 5.7. Uma rede com diversas comunidades: A, B, C, D e E. Todas foram encontradas utilizando a condutância como medida [Leskovec et al., 2008]

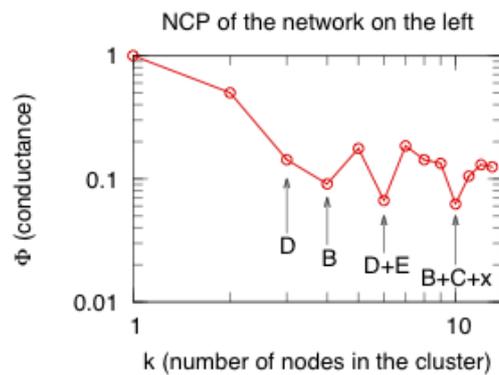


Figura 5.8. Gráfico NCP mostrando a condutância mínima para comunidade de tamanho de 3 a 6 nós. O gráfico se refere ao grafo apresentado na Figura 5.7, as comunidade A, B, C, D e E estão indicadas no gráfico [Leskovec et al., 2008]

por volta de $k = 100$, após esse k o valor da condutância tende a subir. O valor 100 é conhecido como o número de Dunbar, que é o número máximo de relacionamentos que uma pessoa consegue administrar [Dunbar, 1998]. Esta observação sugere que em geral as comunidades não possuem mais do que 100 nós.

Foi observado que o gráfico NCP tende a apresentar um formato em “V”. Este formato pode ser explicado por um formato de comunidade diferente do formato apresentado pela Figura 5.5, observada em diversas redes reais, como [Newman, 2006, Ravasz et al., 2002, Girvan and Newman, 2002]. O formato do gráfico NCP indica que as grandes redes reais apresentam uma estrutura de “Centro-Periferia” aninhada [Borgatti and Everett, 1999, Holme, 2005], que em ciência da computação é conhecida também pelo nome “*jellyfish*” [Tauro et al., 2001] ou “*octopus*” [Chung and Lu, 2006] e que é exemplificada na Figura 5.9. Este conceito significa que uma rede é composta por um grande e denso conjunto de nós (core/centro) ligados entre si que basicamente não tem nenhuma estrutura de comunidade hierárquica, isto é, não podem ser quebrados em

comunidades menores. Assim, a estrutura Centro-Periferia sugere o oposto da estrutura de comunidade hierárquica, e parece ser o mais encontrado em redes complexas de grande escala [Leskovec et al., 2008] e também em redes de computadores chamados Sistemas Autônomos [Siganos, 2006].

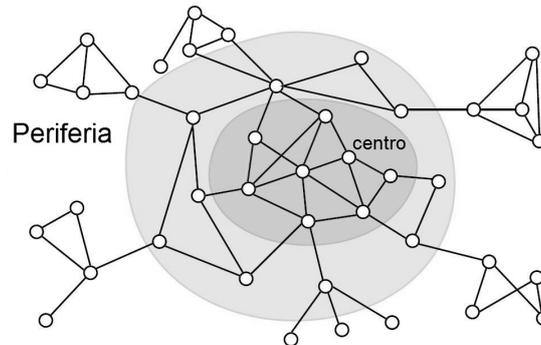


Figura 5.9. Grafo apresentando a topologia “Centro-Periferia”[Leskovec et al., 2008].

A técnica NCP pode ser usada com qualquer outra medida que defina a qualidade da comunidade, não só a condutância. Dentre essas medidas podemos destacar:

- **Espansão:** $f(S) = \frac{cS}{nS}$ - mede o número de arestas por nS que apontam para fora da comunidade [Radicchi et al., 2004].
- **Densidade Interna:** $f(S) = 1 - \frac{mS}{nS(nS-1)}$ - densidade interna das arestas da comunidade S [Radicchi et al., 2004].
- **Corte:** $f(S) = \frac{cS}{nS(n-nS)}$ - fração de todas as arestas possíveis deixando a comunidade [Fortunato, 2010].
- **Corte Normalizado:** $f(S) = \frac{cS}{2mS+cS} + \frac{cS}{2(m-mS)+cS}$ [Shi and Malik, 2000].
- **Maximo-ODF (Out Degree Fraction):** $\max_{u \in S} \frac{|(u,v):v \notin S|}{d(u)}$ - é o fração máxima de arestas de um nó apontando para fora da comunidade [Flake et al., 2000].
- **Média-ODF:** $f(S) = \frac{1}{nS} \sum_{u \in S} \frac{|(u,v):v \notin S|}{d(u)}$ - é a fração de nós média apontando para fora da comunidade [Flake et al., 2000].
- **Flake-ODF:** $f(S) = \frac{|u:u \in S, |(u,v):v \notin S| < d(u)/2|}{nS}$ - é a fração de nós em S que tenha menos arestas apontando para dentro do que para fora da comunidade [Flake et al., 2000].

Os métodos supracitados não permitem a sobreposição de comunidades. A sobreposição é uma característica importante, principalmente nas redes sociais, já que, as pessoas naturalmente participam de mais de um grupo, como, escola, esportes, etc. Assim, um método bastante interessante que permite a sobreposição é o método Cross-Association [Chakrabarti et al., 2004]. Este método faz uma decomposição conjunta da

matriz de adjacência em grupos de linhas e colunas disjuntas, tal que intersecções retangulares são grupos homogêneos. O método é baseado em permutação de linhas e colunas utilizando-se do princípio MDL (Minimum Description Language). A ideia principal é que a matriz binária de uma rede representa a associação entre objetos (linhas e colunas) e quer se encontrar associações cruzadas entre esses objetos, isto é grupos homogêneos retangulares.

A Figura 5.10 apresenta a matriz de adjacência de duas redes reais Epinions e Oregon, processadas por este método. Quanto mais escura a área retangular mais denso é o grupo encontrado. Uma vantagem desse método é que o número de grupos é encontrado pelo método, além disso o número de grupos não precisa ser o mesmo para linhas (K) e colunas (l).

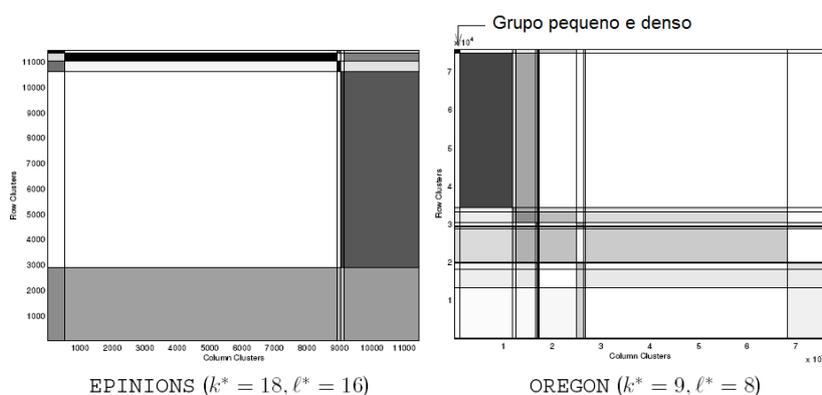


Figura 5.10. Duas redes reais, Epinions e Oregon, processadas pelo método Cross-Association [Chakrabarti et al., 2004].

5.3.3. Predição de Ligações

As redes sociais são objetos altamente dinâmicos, pois eles crescem e mudam rapidamente ao longo do tempo por meio da adição de novas arestas, o que significa o aparecimento de novas interações na estrutura social subjacente. Compreender os mecanismos pelos quais as redes evoluem é uma questão fundamental que ainda não é bem compreendida. A predição de ligações pode ser definida como, dado um “*snapshot*” de uma rede complexa em um tempo t , quer se prever com uma certa acurácia as arestas que irão surgir na rede complexa no tempo futuro $t + 1$.

Com efeito, o problema de predição de ligação aponta para a seguinte pergunta: “até que ponto a evolução de uma rede social pode ser modelada utilizando características intrínsecas à rede em si?” Considere uma rede de co-autoria entre os pesquisadores, por exemplo. Há muitas razões, exógenas à rede, por que dois cientistas que nunca escreveram um artigo juntos vão escrevê-lo nos próximos anos: por exemplo, eles podem se tornarem geograficamente próximos se um deles muda de instituição. Tais colaborações são difíceis de prever. Mas também percebe-se que há um grande número de novas colaborações que são preditas pela topologia da rede: dois pesquisadores que estão “próximos” na rede e tenham inúmeros colegas em comum, e participam dos mesmo círculos sociais sugere que eles são propensos a colaborar em um futuro próximo. O objetivo da predição de ligação é encontrar tais pesquisadores, ou seja de uma maneira mais geral, prever tais ligações.

Uma das dificuldades da predição de ligações é que as redes complexas tendem a ser esparsas. Por exemplo no caso das redes sociais como o Facebook, um usuário típico é conectado com cerca de 100 pessoas sobre um total de 500 milhões de usuários da rede. Para driblar esta dificuldade, alguns modelos fazem uso não só de propriedades estruturais do grafo mas também de características relacionais baseadas nos atributos dos nós do grafo. Esta abordagem é mais conhecida na área de aprendizado relacional ou aprendizado multi-relacional, que tem por objetivo não só o uso da estrutura dos grafos, mas também a descrição dos mesmos por meio de uma base de dados relacional ou lógica relacional ou de primeira ordem. Com isso, o desempenho dos algoritmos pode ser efetivamente melhorado, considerando algumas informações externas, como os atributos dos nós [Getoor and Diehl, 2005, Hasan et al., 2006, Taskar et al., 2004, Popescul et al., 2003]. No senso comum, duas pessoas compartilham mais gostos e interesses (e, portanto, há uma maior probabilidade delas estarem conectados em uma rede social) se elas têm mais características em comum, tais como idade, sexo, trabalho, e assim por diante.

As informações dos atributos podem ser usadas para prever as ligações sem considerar a estrutura de rede. Assim, quando os links existentes não são confiáveis, os métodos baseados em atributos são preferíveis, o que pode, de alguma forma, resolver o chamado problema do começo frio - um grande desafio de previsão link [Leroy et al., 2010]. Além disso, a estrutura de comunidade também pode ajudar a melhorar a precisão da previsão [Zheleva et al., 2010]. Em redes sociais, uma vez que uma pessoa pode desempenhar diferentes papéis em diferentes comunidades, a previsão em um domínio pode ser inspirada pelas informações de outros [Cao et al., 2010]. Por exemplo, quando as colaborações entre os autores são previstas, pode-se considerar suas filiações para melhorar a precisão. Entretanto, esta informação complementar dos nós e arestas nem sempre estão disponíveis, o que inviabiliza a aplicação desses algoritmos nesses casos.

Dentre as técnicas de predição de ligação destacam-se as baseadas em propriedades estruturais do grafo [Liben-Nowell and Kleinberg, 2003, Huang, 2006], especialmente as que trabalham de maneira local na rede. Essas medidas tem como técnica principal atribuir um valor de ligação, chamado $score(u; w)$, para pares de nós $\langle u, w \rangle$ baseado em um dado grafo \mathcal{G} . Os valores atribuídos são ordenados em ordem decrescente e então as predições são feitas de acordo com esta lista. Os valores computados podem ser vistos como medidas de proximidades entre nós u e w , relativos a topologia da rede. Contudo, qualquer tipo de medida que compute a semelhança entre dois nós pode ser utilizada. Uma boa revisão na área pode ser encontrada em [Lu and Zhou, 2010].

Para o nó u , seja $\Gamma(u)$ o número de vizinhos de u em \mathcal{G} . Um grande número de técnicas de predição de ligações são baseadas na ideia que dois nós u e w devem possuir uma aresta entre eles no futuro, se o conjunto dos seus vizinhos $\Gamma(u)$ e $\Gamma(w)$ tiverem uma grande sobreposição. Esta técnica segue a intuição natural que, em uma rede de co-autoria entre os pesquisadores supracitado, por exemplo, os nós u e w representam autores e $\Gamma(u)$ e $\Gamma(w)$ o conjunto de colegas com quem u e w tiveram pelo menos uma publicação em comum. Assim, quanto maior o número $\Gamma(u) \cap \Gamma(w)$ maior o número de colegas em comum u e w compartilha e consequentemente maior a probabilidade de u e w se tornarem colegas no futuro.

A implementação mais direta desta técnica de predição de ligação é chamada

“vizinhos-comum”, sobre o qual para cada par de nó distante por duas arestas e não conectados diretamente é definido um valor que representa o número de vizinhos comuns que u e w compartilham. A Equação 2 apresenta esta medida.

$$\text{score}(u; w) := |\Gamma(u) \cap \Gamma(w)| \quad (2)$$

A técnica “vizinhos-comum” captura a noção de que dois estranhos que possuem um amigo em comum podem ser apresentados por este amigo. Isto introduz o efeito de “triângulo fechado” em um grafo e se assemelha a um mecanismo da vida real como apresentado na Figura 5.11.

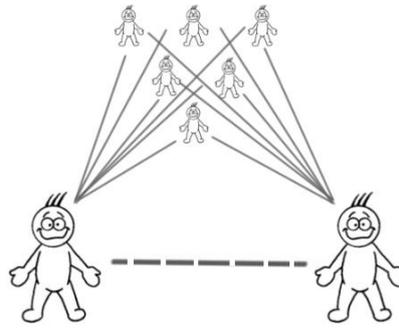


Figura 5.11. A conexão entre dois nós (pessoas) que compartilham muitos vizinhos.

Um exemplo clássico deste mecanismo é a indicação de possíveis amigos em redes sociais. Em muitas redes, especialmente as redes sociais, é notado que se um nó u é conectado com um nó v que é conectado com w , então há uma grande probabilidade de u ser conectado com w . Esta relação é chamada de transitividade e é medida pelo coeficiente de clusterização [Watts and Strogatz, 1998]. A transitividade significa a presença de um alto número de triângulos ($\Delta(v_i)$) na rede. A contagem de triângulos é a principal parte do coeficiente de clusterização, que pode ser calculado para cada nó do grafo (Equação 3) ou para o grafo como um todo (Equação 4). Este coeficiente tem o objetivo de indicar quão próximo o grafo está de ser um grafo completo. O coeficiente de clusterização $C(v_i)$ de um nó v_i de grau $d(v_i)$ é definido pela Equação 3 a seguir.

$$C(v_i) = \frac{2 * \Delta(v_i)}{d(v_i) * (d(v_i) - 1)} \quad (3)$$

Seja v_i um nó com grau $|d(v_i)|$, então no máximo $d(v_i) * (d(v_i) - 1) / 2$ arestas podem existir entre eles, sendo $\Delta(v_i)$ a fração de arestas que realmente existe, isto é o número de triângulos. Isto significa que, o coeficiente de clusterização $C(v_i)$ de um nó v_i é a proporção de arestas entre os nós da sua adjacência dividido pelo número de arestas que podem existir entre eles. Equivalentemente, $C(v_i)$ é a fração de triângulos centrados no nó v_i entre $(d(v_i) * (d(v_i) - 1)) / 2$ triângulos que possam existir.

O coeficiente de clusterização global $C(\mathcal{G})$ é a média da soma de todos os $C(v_i)$ dos nós do grafo \mathcal{G} , dividido pelo número total de nós N . A equação do coeficiente de

clusterização global é apresentada na Equação 4 a seguir.

$$C(\mathcal{G}) = \frac{1}{N} * \sum_{i=1}^N C(v_i) \quad (4)$$

Em Newman [Newman, 2001] foi computado esta quantidade ($\mathfrak{K}(u, w)$) no contexto da rede de colaboração, verificando uma correlação positiva entre o número de vizinhos comuns de u e w no tempo t , e a probabilidade de u e w colaborarem em algum período depois de t .

O coeficiente de Jaccard [Salton and McGill, 1983] é uma similaridade métrica que é comumente utilizada na recuperação de informação. No contexto da predição de ligação ela é utilizada para medir a probabilidade de ambos u e w ter a característica f , para uma característica f selecionada randomicamente dentre as características em comum entre u e w . Se a característica f escolhida for o número de vizinhos em comum em G , então esta medida captura intuitivamente a noção de proporção de vizinhos de u que também são vizinhos de w (e vice-versa) o que é uma boa medida de similaridade de u e w . Formalmente, o coeficiente de Jaccard usa a Equação 5.

$$score(u; w) := \frac{|\Gamma(u) \cap \Gamma(w)|}{|\Gamma(u) \cup \Gamma(w)|} \quad (5)$$

Adamic e Adar é uma medida semelhante ao coeficiente de Jaccard, no contexto de decidir se dois nós são fortemente relacionados ou não. Para este cálculo, ele computa as características dos nós e defini a similaridade entre eles como apresenta a Equação 6.

$$score(u; w) := \sum_{v \in \Gamma(u) \cap \Gamma(w)} \frac{1}{\log \Gamma(v)} \quad (6)$$

A medida Adamic/Adar [Adamic and Adar, 2003] avalia o grau dos vizinhos comum e dá ênfase aos nós em que os vizinhos em comum possuem um grau baixo. Isto por que nós com alto grau tem uma maior chance de ser vizinhos de muitos nós. Assim, se duas pessoas têm em comum um amigo que possui poucos amigos, estas duas pessoas tem uma probabilidade maior de serem amigas em um futuro do que duas pessoas que tem como amigo em comum uma pessoa muito popular que tem inúmeros amigos.

Em suma, se o fechamento de triângulo é um mecanismo pelo qual novas arestas são adicionadas em uma rede, então para u e w serem apresentados por um amigo em comum v , a pessoa v terá que escolher introduzir o par $\langle u; w \rangle$ de $\left(\frac{\Gamma(v)}{2}\right)$ pares de amigos, assim u e w têm mais chance de serem apresentados se v for uma pessoa não popular já que a quantidade de pares de amigos será menor.

Apesar da grande quantidade de medidas desenvolvidas na predição de ligação, os resultados apresentados em [Zhou et al., 2009] indicam que a medida mais simples, “vizinhos-comum”, possui a melhor performance sobre as outras medidas. Além de ser uma medida de fácil cálculo e fácil adaptação a outros problemas especialmente quando se utiliza classes de nós.

Uma outra medida ainda local desenvolvida é a “Acoplamento Preferencial” (*Preferential attachment*). Esta medida recebeu bastante atenção com o desenvolvimento do modelo de crescimento de uma rede chamado *Preferential attachment*. A primícia básica desta medida é que a nova aresta é anexa a u proporcionalmente ao valor de $\Gamma(u)$. Newman e Barabasi propuseram, baseado em experimentos empíricos, que a probabilidade de u e w estarem correlacionados é corresponde a Equação 7.

$$score(u; w) := \Gamma(u) \cdot \Gamma(w) \quad (7)$$

Além dessas medidas baseadas em vizinhança também há medidas baseadas na contagem de caminhos. Um grande número de métodos refinam a noção de distância de caminho mínimo para a consideração implícita de utilizar todos os caminhos entre dois nós. Dentre esta abordagem se encontra a técnica *Katz* que define a medida como sendo a soma direta sobre toda a coleção de caminhos, exponencialmente amortecida para contabilizar caminhos mais curtos de maneira mais forte como apresenta a Equação 8.

$$score(u; w) := \sum_{l=1}^{\infty} \beta^l \cdot |path_{u,w}^{(l)}| \quad (8)$$

Sendo $path_{u,w}^{(l)}$ todos os caminhos de tamanho l entre u e w . Usando um β muito pequeno esta medida se aproxima da medida “vizinhos-comum” já que caminhos de comprimento maior que 3 contabilizam muito pouco na soma. Para esta medida há variantes para grafos com peso e sem peso. Sem peso $path_{u,w}^{(1)} = 1$ se há um caminho entre u e w , caso contrário é 0. Com peso o valor de $path_{u,w}^{(1)}$ é o número de caminhos ou o peso da aresta entre u e w .

Outra medida comumente utilizada baseada em caminhos é a *random walk*. Uma *random walk* no grafo \mathcal{G} começa no nó u e se move iterativamente para a vizinhança de u que é escolhida de maneira randômica e uniforme. O tempo de alcance $H_{u,w}$ de u para w é o número esperado de passos requerido para uma *random walk* começando em u alcançar w . Um exemplo eficiente dessa abordagem é feito em [Backstrom and Leskovec, 2011] na qual foi desenvolvido o método de *Random Walks* Supervisionada que naturalmente combina informações estruturais da rede com atributos de nós e arestas. Esses atributos são utilizados para guiar a *random walk*. A tarefa supervisionada é usada para aprender funções e atribuir peso às arestas da rede de maneira que nós que terão uma aresta criada no futuro tem mais probabilidades de ser visitados pela *random walk*.

Um exemplo interessante é apresentado em [Clauset et al., 2008], em que a descoberta de grupos, isto é comunidades, em redes complexas é usado para o auxílio na identificação de ligações faltantes, já que pares de nós pertencentes a uma mesma comunidade têm mais chance de serem conexos entre si do que pares de nós pertencentes a comunidades diferentes. Este método se diferencia da predição de ligação tradicional, pois normalmente esta tarefa visa descobrir arestas que virão a existir na rede complexa quando esta evoluir (crescimento do número de nós e arestas com o passar do tempo) e não uma aresta perdida na construção da rede complexa. Entretanto, este método não funciona para todos os tipos de redes complexas, já que o método não consegue detec-

tar comunidades em redes complexas que não possuam grupos bem definidos. Há uma coleção de trabalhos que vem usando e desenvolvendo algoritmos na área de predição de ligações [Kashima et al., 2009, Hasan et al., 2006, Kunegis and Lommatzsch, 2009, Lu and Zhou, 2009, Acar et al., 2009].

Em muitos domínios identificar ligações anômalas pode ser mais útil que predizer ligações [Rattigan and Jensen, 2005]. A descoberta de ligações anômalas é uma tarefa dependente da predição de ligação, já que é baseada nas mesmas técnicas mas para encontrar arestas com comportamento suspeito ao invés de identificar arestas que aparecerão em um futuro próximo. Exemplos de aplicações estão descoberta de usuários fantasmas em redes sociais, cartões de créditos, aprendizado errôneo em ontologias, etc.

A significativa contribuição do estudo de redes complexas com a predição de ligação é o conhecimento profundo sobre os fatores estruturais que afetam o desempenho de algoritmos, que também pode ser considerado como a orientação da escolha dos algoritmos quando ambos a precisão e complexidade tem que ser levada em conta. Por exemplo, se a rede é altamente clusterizada, os algoritmos baseado em vizinhança comum podem ser boas escolhas, uma vez que pode dar uma previsão relativamente boa com uma complexidade muito baixa. No entanto, se a rede não é altamente clusterizada, ou a distribuição do número de vizinhos comuns decai muito rápido (como no nível de roteamento da Internet [56], 99,98% dos pares de nós não compartilham mais de dois vizinhos comum), algoritmos baseado em vizinhança comum são muito pobres, e devemos tentar abordagens baseada em caminho e *random walk* que fazem uso de mais informações.

Até agora a predição de ligação tem focado redes não direcionadas e sem pesos. Para redes direcionadas, encontrar triângulos é uma tarefa mais complexa fazendo com que mesmo a medida mais simples que é a de vizinhança comum necessite ser modificada para ser utilizada em tais redes. Nesta medida, mesmo que encontrar uma aresta seja uma modificação fácil, definir a direção da aresta pode ser uma tarefa complexa [Mantrach et al., 2009]. A maneira correta de explorar as informações de pesos para melhorar a precisão da predição ainda é um problema não resolvido. Um problema mais difícil é prever os pesos das ligações, o que é relevante para a previsão de tráfego para os sistemas de transporte urbano e transporte aéreo [Yin et al., 2002, Murata and Moriyasu, 2007, Lu and Zhou, 2009, Lu and Zhou, 2010]

Um grande desafio é a predição de ligação em redes multi-dimensionais, onde as ligações podem ter significados diferentes. Por exemplo, uma rede social pode consistir de links positivos e negativos, respectivamente, apontando para amigos e inimigos [Kunegis et al., 2009], ou de confiança e desconfiança [Guha et al., 2004]. Em [Leskovec et al., 2010] propôs-se um método para prever os sinais de links (positivo ou negativo), mas a previsão de tanto a existência de um link e seu sinal não foi bem estudada ainda. Desenvolvimento recente da teoria do equilíbrio social, podem oferecer sugestões úteis [Traag and Bruggeman, 2009, Marvel et al., 2009, Szell et al., 2010].

5.4. Conclusão

Este documento apresentou uma visão geral da área de detecção de comunidades e predição de ligação dentro da mineração de grafos e redes complexas. Esta área tem se

mostrado muito importante atualmente, principalmente pelo grande crescimento do domínio de aplicação, nos quais os dados podem ser modelados através de redes complexas com especial atenção às redes sociais.

Apesar das origens remotas e da grande popularidade dos últimos anos, a pesquisas em detecção de comunidades ainda não forneceu uma solução satisfatória ao problema e deixa com uma série de importantes questões em aberto. O campo não tem uma abordagem teórica que define precisamente o que algoritmos de detecção de comunidade deve fazer, porém, todos têm a sua própria ideia do que é uma comunidade, e mais, as ideias são consistentes entre si, mas, desde que ainda há discordância, continua a ser impossível decidir qual algoritmo faz o melhor trabalho e não há nenhum controle sobre a criação de novos métodos. Portanto, em primeiro lugar a comunidade científica que trabalha em detecção de comunidades deve definir um conjunto de grafos de referência os quais devem servir de base para novos algoritmos.

A predição de ligação apesar de estar melhor definida que a detecção de comunidade ainda possui também o longo caminho para alcançar a maturidade. Sendo a principal questão atender a diversos tipos de redes complexas, como as direcionadas e com pesos e também prever não só a falta de ligação mas também a ligação errônea.

O documento apresentou ainda definições da teoria dos grafos necessárias para o entendimento desses algoritmos e propriedades básicas também foram abordadas. As redes complexas estão cada vez mais presentes nos sistemas computacionais e com isso o seu entendimento torna-se cada vez mais importante e relevante tanto para pesquisadores da área da computação quanto para de áreas em que problemas reais podem ser representadas através destes modelos. No campo na mineração de redes complexas, especialmente para a detecção de comunidades e a predição de ligações há muito trabalho a ser feito para estas tarefas atinjam a maturidade semelhante a tarefas na área de mineração tradicional como regras de associação e clusterização.

Agradecimentos

Os autores agradecem ao CNPq pelo financiamento.

Referências

- [Acar et al., 2009] Acar, E., Dunlavy, D. M., and Kolda, T. G. (2009). Link prediction on evolving data using matrix and tensor factorizations. In Saygin, Y., Yu, J. X., Kargupta, H., Wang, W., Ranka, S., Yu, P. S., and Wu, X., editors, *ICDM Workshops*, pages 262–269. IEEE Computer Society.
- [Adamic and Adar, 2003] Adamic, L. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211–230.
- [Alon, 1998] Alon, N. (1998). Spectral techniques in graph algorithms. In Lucchesi, C. L. and Moura, A. V., editors, *Lecture Notes in Computer Science 1380*, pages 206–215. Springer-Verlag, Berlin.
- [Backstrom and Leskovec, 2011] Backstrom, L. and Leskovec, J. (2011). Supervised random walks: predicting and recommending links in social networks. In King, I.,

- Nejdl, W., and Li, H., editors, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 635–644. ACM.
- [Barabasi, 2002] Barabasi, A.-L. (2002). *Linked: The New Science of Networks*. Perseus Publishing, 1st edition.
- [Bondy and Murty, 1979] Bondy, J. A. and Murty, U. S. R. (1979). *Graph Theory with applications*. Elsevier Science Publishing Co., Inc.
- [Borgatti and Everett, 1999] Borgatti, S. P. and Everett, M. G. (1999). Models of core/periphery structures. *Social Networks*, 21.
- [Brandes and Erlebach, 2005] Brandes, U. and Erlebach, T. (2005). *Network Analysis: Methodological Foundations*. Springer.
- [Cao et al., 2010] Cao, B., Liu, N. N., and Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. In *International Conference on Machine Learning*, pages 159–166.
- [Chakrabarti et al., 2004] Chakrabarti, D., Papadimitriou, S., Modha, D. S., and Faloutsos, C. (2004). Fully automatic cross-associations. In *KDD '04: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 79–88. ACM Press.
- [Chung and Lu, 2006] Chung, F. and Lu, L. (2006). *Complex Graphs and Networks*. American Mathematical Society.
- [Clauset et al., 2008] Clauset, A., Moore, C., and Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101.
- [Clauset et al., 2009] Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–704.
- [Coleman, 1988] Coleman, J. S. (1988). Social capital in the creation of human capital. *American Journal of Sociology*, 94.
- [Diestel, 2005] Diestel, R. (2005). *Graph Theory*. Springer-Verlag Heidelbergh.
- [Dunbar, 1998] Dunbar, R. (1998). *Grooming, Gossip, and the Evolution of Language*. Harvard Univ Press.
- [Faloutsos et al., 1999] Faloutsos, M., Faloutsos, P., and Faloutsos, C. (1999). On power-law relationships of the internet topology. In *SIGCOMM 1999*, volume 1, pages 251–262, Cambridge, Massachusetts. ACM Press.
- [Flake et al., 2000] Flake, G. W., Lawrence, S., and Giles, C. L. (2000). Efficient identification of Web communities. In *KDD*.
- [Flake et al., 2002] Flake, G. W., Lawrence, S., Giles, C. L., and Coetzee, F. M. (2002). Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71.

- [Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75–174.
- [Freeman, 2004] Freeman, L. C. (2004). *The Development of Social Network Analysis: A Study in the Sociology of Science*. Empirical Press.
- [Getoor and Diehl, 2005] Getoor, L. and Diehl, C. P. (2005). Introduction to the special issue on link mining. *SIGKDD Explor. Newsl.*, 7(2):1–2.
- [Girvan and Newman, 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. In *Proc. Natl. Acad. Sci. USA*, volume 99.
- [Guha et al., 2004] Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 403–412, New York, NY, USA. ACM.
- [Hasan et al., 2006] Hasan, M. A., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*.
- [Holme, 2005] Holme, P. (2005). Core-periphery organization of complex networks. *Phys. Rev. E*, 72(4):046111.
- [Huang, 2006] Huang, Z. (2006). Link prediction based on graph topology: The predictive value of the generalized clustering coefficient. In *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (LinkKDD2006)*.
- [Jeong et al., 2000] Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, A.-L. (2000). The large-scale organization of metabolic networks. *Nature*, 407.
- [Kannan et al., 2000] Kannan, R., Vempala, S., and Vetta, A. (2000). On clusterings – good, bad and spectral. In *FOCS*.
- [Karypis and Kumar, 1998] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392.
- [Kashima et al., 2009] Kashima, H., Kato, T., Yamanishi, Y., Sugiyama, M., and Tsuda, K. (2009). Link propagation: A fast semi-supervised learning algorithm for link prediction. In *SDM*, pages 1099–1110. SIAM.
- [Krishnamurthy and Wang, 2000] Krishnamurthy, B. and Wang, J. (2000). On network-aware clustering of web clients. *SIGCOMM Comput. Commun. Rev.*, 30(4):97–110.
- [Kunegis and Lommatzsch, 2009] Kunegis, J. and Lommatzsch, A. (2009). Learning spectral graph transformations for link prediction. In *Proc. Int. Conf. in Machine Learning*.
- [Kunegis et al., 2009] Kunegis, J., Lommatzsch, A., and Bauckhage, C. (2009). The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web, WWW '09*, pages 741–750, New York, NY, USA. ACM.

- [Leroy et al., 2010] Leroy, V., Cambazoglu, B. B., and Bonchi, F. (2010). Cold start link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 393–402, New York, NY, USA. ACM.
- [Leskovec et al., 2010] Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 641–650, New York, NY, USA. ACM.
- [Leskovec et al., 2007] Leskovec, J., Kleinberg, J. M., and Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM TKDD*, 1(1):1 – 40.
- [Leskovec et al., 2008] Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2008). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR*, abs/0810.1355.
- [Liben-Nowell and Kleinberg, 2003] Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA. ACM.
- [Liu and Wong, 2008] Liu, G. and Wong, L. (2008). Effective pruning techniques for mining quasi-cliques. In *ECML PKDD '08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, pages 33–49, Berlin, Heidelberg. Springer-Verlag.
- [Lu and Zhou, 2009] Lu, L. and Zhou, T. (2009). Role of weak ties in link prediction of complex networks. In *Proceedings of the 1st ACM International Workshop on Complex Networks in Information and Knowledge Management (CNIKM)*, Hong Kong, China.
- [Lu and Zhou, 2010] Lu, L. and Zhou, T. (2010). Link prediction in complex networks: A survey.
- [Mancoridis et al., 1998] Mancoridis, S., Mitchell, B. S., and Rorres, C. (1998). Using automatic clustering to produce high-level system organizations of source code. In *In Proc. 6th Intl. Workshop on Program Comprehension*, pages 45–53.
- [Mantrach et al., 2009] Mantrach, A., Yen, L., Callut, J., Francois, K., Shimbo, M., and Saerens, M. (2009). The sum-over-paths covariance kernel: A novel covariance measure between nodes of a directed graph. In *the IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [Marvel et al., 2009] Marvel, S. A., Strogatz, S. H., and Kleinberg, J. M. (2009). The energy landscape of social balance.
- [Murata and Moriyasu, 2007] Murata, T. and Moriyasu, S. (2007). Link prediction of social networks based on weighted proximity measures. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 85–88, Washington, DC, USA. IEEE Computer Society.

- [Newman, 2010] Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.
- [Newman, 2001] Newman, M. E. J. (2001). Scientific collaboration networks: II. shortest paths, weighted networks, and centrality. *Physics Review E*, 69.
- [Newman, 2005] Newman, M. E. J. (2005). Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46:323.
- [Newman, 2006] Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.
- [Newman and Girvan, 2003] Newman, M. E. J. and Girvan, M. (2003). Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113+.
- [Nicoletti, 2006] Nicoletti, Maria Do Carmo ; Hruschka Jr., E. (2006). *Fundamentos da Teoria dos Grafos.*, volume 1. EdUFSCar - Editora da Universidade Federal de São Carlos, 1. ed. revisada edition.
- [Palla et al., 2005] Palla, G., Derenyi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.
- [Popescul et al., 2003] Popescul, A., Popescul, R., and Ungar, L. H. (2003). Statistical relational learning for link prediction.
- [Radicchi et al., 2004] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663.
- [Rattigan and Jensen, 2005] Rattigan, M. J. and Jensen, D. (2005). The case for anomalous link detection. In *Proceedings of the 4th international workshop on Multi-relational mining*, MRDM '05, pages 69–74, New York, NY, USA. ACM.
- [Ravasz et al., 2002] Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabasi, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555.
- [Reddy et al., 2002] Reddy, P. K., Kitsuregawa, M., Sreekanth, P., and 0002, S. S. R. (2002). A graph based approach to extract a neighborhood customer community for collaborative filtering. In *DNIS*, pages 188–200.
- [Redner, 1998] Redner, S. (1998). How popular is your paper? an empirical study of the citation distribution.
- [Salton and McGill, 1983] Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.
- [Schaeffer, 2007] Schaeffer, S. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

- [Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:888–905.
- [Siganos, 2006] Siganos, Georgos; Sudhir L Tauro, M. F. (2006). Jellyfish: A conceptual model for the as internet topology. In *Journal of Communications and Networks*, volume 8, pages 339 – 350.
- [Szell et al., 2010] Szell, M., Lambiotte, R., and Thurner, S. (2010). Multirelational organization of large-scale social networks in an online world.
- [Taskar et al., 2004] Taskar, B., Wong, M., Abbeel, P., and Koller, D. (2004). Link prediction in relational data.
- [Tauro et al., 2001] Tauro, S. L., Palmer, C., Siganos, G., and Faloutsos, M. (2001). A simple conceptual model for the internet topology. In *Global Internet, San Antonio, Texas*.
- [Traag and Bruggeman, 2009] Traag, V. A. and Bruggeman, J. (2009). Community detection in networks with positive and negative links. *Physical Review E*, 80(3).
- [Tsourakakis, 2008] Tsourakakis, C. E. (2008). Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM '08*, pages 608–617, Washington, DC, USA. IEEE Computer Society.
- [von Luxburg, 2007] von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.
- [Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of small-world networks. *Nature*, 393(6684):440–442.
- [Yin et al., 2002] Yin, H., Wong, S. C., Xu, J., and Wong, C. K. (2002). Urban traffic flow prediction using a fuzzy-neural approach. *Transportation Research Part C: Emerging Technologies*, 10(2):85 – 98.
- [Zheleva et al., 2010] Zheleva, E., Getoor, L., Golbeck, J., and Kuter, U. (2010). Using friendship ties and family circles for link prediction. In *Proceedings of the Second international conference on Advances in social network mining and analysis, SNAKDD'08*, pages 97–113, Berlin, Heidelberg. Springer-Verlag.
- [Zhou et al., 2009] Zhou, T., Lü, L., and Zhang, Y.-C. (2009). Predicting missing links via local information. *The European Physical Journal B - Condensed Matter and Complex Systems*, 71(4):623–630.

Chapter

6

Utilizando a Arthron para o Gerenciamento Remoto de Múltiplos Fluxos de Mídia

Julio César Ferreira da Silva, Anderson Vinícius Alves Ferreira, Elenilson Vieira da Silva Filho, Marcello Galdino Passos, Erick Augusto Gomes de Melo, Tatiana Aires Tavares e Guido Lemos de Souza Filho

LAViD – Laboratório de Aplicações de Vídeo Digital – Universidade Federal da Paraíba (UFPB) {julio@lavid.ufpb.br, anderson@lavid.ufpb.br, elenilson@lavid.ufpb.br, marcello@lavid.ufpb.br, erick@lavid.ufpb.br, tatiana@lavid.ufpb.br, guido@lavid.ufpb.br}

Abstract

The use of technology to support the transmission of media streams, such as those generated in artistic and technological, sports and techno-scientific events or in telemedicine, makes Human Computer Interaction more sophisticated, natural and complex, especially from the point of view of computer systems. Thus, this short course aims to present Arthron, a tool consisting of a components set that together facilitates distributed streams management controlling everything efficiently and intuitively, from media capture to media display.

Resumo

O uso da tecnologia para dar suporte à transmissão de fluxos de mídia, como os que são gerados em eventos artístico-tecnológicos, tecno-científicos e esportivos ou em telemedicina, torna a Interação Humano Computador mais sofisticada, natural e complexa, especialmente do ponto de vista dos sistemas computacionais. Assim, este minicurso tem como objetivo apresentar a Arthron, uma ferramenta composta por um conjunto de componentes que juntos facilitam o gerenciamento de fluxos distribuídos, controlando desde a captura até a exibição das mídias envolvidas de forma eficiente e intuitiva.

6.1. Apresentação

O avanço rápido e contínuo das Tecnologias de Informação e Comunicação (TICs) impulsionou o desenvolvimento das “super redes” de computadores tornando possível o envio e recebimento de grandes volumes de dados. Neste novo cenário, destacam-se os serviços que envolvem vídeo digital e, especialmente, vídeo digital em alta definição. A alta definição traz consigo novas possibilidades de interação, as quais aproximam o observador da fonte em exibição de tal forma que a sensação é o compartilhamento de um mesmo espaço, virtual ou real. São ilustrados, a seguir, alguns exemplos em que a distribuição de conteúdo digital pode ser aplicada.

A aproximação entre Arte e Tecnologia já não é mais uma tendência, mas sim uma realidade [Santana 2003]. A Arte (qualquer arte, de qualquer tempo) sempre foi produzida com os meios tecnológicos de seu tempo. É a apropriação que ela faz do aparato tecnológico que lhe é contemporâneo. Em geral, aparelhos, instrumentos e máquinas semióticas não são projetados para a produção de arte, pelo menos não no sentido secular desse termo, tal como ele se constituiu no mundo moderno, aproximadamente a partir do século XV. No entanto, no cenário de convergência que se vivencia atualmente, a sociedade está em meio a uma metamorfose rumo a um patamar tecnológico que é culturalmente composto por imagens, sons, textos, vídeos e outras mídias.

Nos campos da Telemedicina e Telessaúde, as ações voltadas para a utilização de conteúdo digital vêm crescendo num ritmo acelerado. Grandes empresas de tecnologia como Polycom, Tandberg e Cisco estão investindo pesado nestes campos. A Cisco, por exemplo, apresentou na Conferência de 2010 da Healthcare Information and Management Systems Society (HIMSS) a Cisco HealthPresence, uma nova tecnologia de Telemedicina avançada que permite uma ligação remota entre médicos e clínicos para consultas médicas, com funcionalidades e tecnologias nunca antes utilizadas. Combinando vídeo em alta definição, excelente qualidade de áudio, e permitindo a transmissão de dados médicos, dando a sensação ao paciente de estar participando de uma consulta presencial.

Neste contexto a transmissão de fluxos de vídeo de qualidade variada é uma necessidade cada vez mais presente no dia-a-dia de empresas, instituições de ensino e outras organizações que possam usufruir desses benefícios. No entanto, o universo de soluções dedicadas ou proprietárias é usualmente o mais comum nesse nicho de aplicação. Neste capítulo vamos apresentar a utilização da Arthron, uma ferramenta livre, voltada para a transmissão, codificação e o gerenciamento de múltiplos fluxos de mídia.

6.2 Conceitos e definições

Alguns conceitos são amplamente utilizados durante o processo de desenvolvimento de aplicações baseadas em vídeo digital. Tais conceitos são detalhados nas subseções seguintes.

6.2.1 Captura

A captura de conteúdo multimídia consiste basicamente na conversão de um sinal analógico, a exemplo do sinal produzido por uma câmera de vídeo ou de uma guitarra, para um formato digital. O produto dessa conversão é referenciado por *stream* (fluxo)

de mídia e pode ser utilizado pela aplicação. Por exemplo, um conjunto de imagens exibidas com o passar do tempo é a maneira mais comum de representar vídeo. Toda representação analógica dessas imagens tem sua analogia digital em um formato específico. A princípio, as informações necessárias para gerar um vídeo são suas dimensões (número de linhas e colunas) e a taxa de quadros. A captura de vídeo digital consiste em transformar imagens analógicas obtidas a partir de um dispositivo como câmeras de vídeo em conteúdo digital para que possam ser utilizadas pela aplicação [Morris 2000].

6.2.2 Codificação

Codificação é o conceito de reduzir o volume de dados multimídia sem perder a qualidade percepção humana da informação. Essa redução pode ser alcançada de duas maneiras: a primeira é por remover a redundância, descartando a informação repetida e a segunda é por descartar a informação perceptivelmente insignificante. A codificação de áudio, vídeo ou imagens é necessária considerando o volume de dados necessário para armazenar uma amostra. Pode-se alistar dentre os formatos de codificação largamente conhecidos os seguintes: de áudio o MP3 e o AAC, de vídeo MPEG-2 e o WMV e de imagens JPEG e PNG. Codificar e decodificar dados são processos computacionalmente intensos. Historicamente, quando processadores tinham capacidades limitadas, hardware adicional de tarefa específica era necessário para codificar e decodificar mídia. Hoje, porém, com o aumento do poder de processamento, softwares especializados podem desempenhar essa função [Morris 2000].

6.2.3 Streaming

Streaming é a entrega de conteúdo multimídia capturado diretamente da fonte para um player em tempo-real. É um processo contínuo sem intermediários de armazenamento. Se o conteúdo tiver sido armazenado em um processo de entrega sob demanda, essa entrega deve ter a taxa controlada para a reprodução em tempo-real [Austerberry 2005]. Streaming de mídia é fortemente baseado em protocolos de transmissão (UDP, RTP) e formatos de arquivos (MPEG-4, WAV).

6.2.4 Replicação

A distribuição do conteúdo multimídia entre componentes do sistema é de importância crítica. Uma abordagem frequentemente abordada é a replicação. Replicação consiste em realizar, em um cluster de servidores multimídia, clones de porções do conteúdo gerado e distribuí-los entre os componentes ciclicamente ou segundo alguma política de distribuição pré-estabelecida. Cada requisição de clientes é enviada ao servidor com menor carga, que, então, distribui o conteúdo para o cliente. O custo em termos de tempo para efetuar a replicação cresce proporcionalmente à quantidade de destinos do agente replicador [Dukes 2004].

6.3 A Arthron

Nesta seção apresentamos a Arthron [Melo 2010] ferramenta desenvolvida pelo LAVID/UFPB e RNP para permitir a transmissão e manipulação de múltiplos streamings de mídia remotamente.

A Arthron é uma ferramenta de gerenciamento remoto para captura e distribuição de múltiplos fluxos simultâneos de mídia a fim de prover suporte a diversos cenários de videocolaboração. A Arthron tem por principal funcionalidade oferecer ao usuário uma interface simples para manipulação de diferentes fontes/fluxos de mídia simultâneos. Dessa forma o usuário pode remotamente adicionar, remover, configurar o formato de apresentação e programar a exibição no tempo e no espaço dos fluxos de mídia.

Comparada a diversos trabalhos na literatura que têm como objetivo principal fazer a manipulação de mídias, em especial, de vídeo digital, como o Grass Valley, o SuperCollider, o DICE, o INTERACT e SLTV, temos a seguinte análise comparativa:

Tabela 1: Análise Comparativa

	Suporte TCP/IP	Streaming de Áudio	Streaming de Vídeo	Monitoramento	Distribuição
Arthron	Sim	Sim	Sim	Sim	Sim
Grass Valley	Não	Sim (apenas local)	Sim (apenas local)	Não	Não
SuperCollider	Sim (apenas áudio)	Sim	Não	Não	Sim
DICE	Não	Não	Sim (apenas local)	Não	Não
INTERACT	Não	Sim (apenas local)	Sim (apenas local)	Sim	Não
SLTV	Sim	Sim	Sim	Não	Sim

6.3.1 Arquitetura

Desde o início do projeto de desenvolvimento da Arthron, a evolução da arquitetura acomodou as novas características geradas durante o desenvolvimento da Arthron, uma vez que foi utilizado um modelo iterativo de desenvolvimento. A arquitetura inicial da Arthron apresentava os componentes Articulator, Decoder, Reflector, Encoder e VideoServer. O componente Reflector é responsável pela replicação e redistribuição das mídias. A utilização desse componente também provocou um melhor aproveitamento do uso da rede, pois permitiu inserir um intermediário na retransmissão do fluxo melhorando assim a utilização da rede.

A Figura 1 apresenta a versão atual da arquitetura, que possui dois novos componentes: o VideoRoom e o WebService. Estes componentes surgiram da necessidade de inserir novas funcionalidades na Arthron. Por exemplo, o componente VideoRoom surgiu devido à necessidade de acomodar as funcionalidades de Decoders e um Encoder em um mesmo componente (como será detalhado a seguir). O componente WebService surgiu da necessidade de inserir mecanismos de garantia de segurança no provimento dos serviços no contexto de Telemedicina.

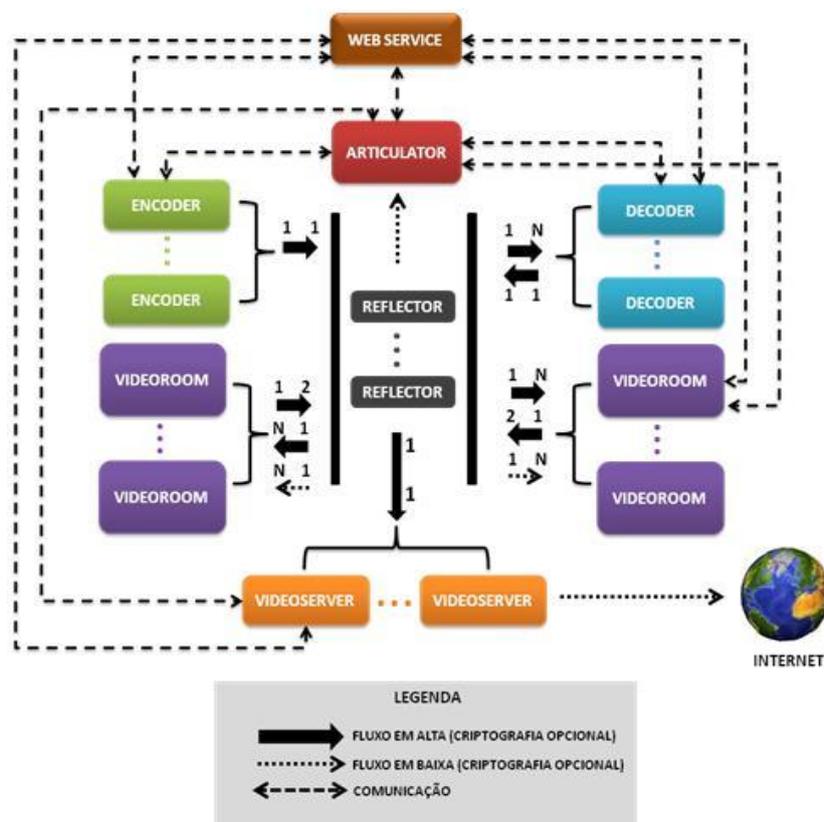


Figura 1: Arquitetura atual da Arthron

A ferramenta é composta pelos seguintes componentes: Articulador (Manager), agente codificador (Encoder), agente decodificador (Decoder), distribuidor de fluxo (Refletor), Servidor de Vídeo (VideoServer), Servidor de gerenciamento do usuário (WebService) e o Agente para Videoconferência (VideoRoom). Esses fluxos podem ser gerados em localizações geograficamente distribuídas e são manipulados pela ferramenta Arthron que é responsável por capturar, controlar, transcodificar, transmitir e decodificar as mídias capturadas em tempo real ou de arquivo. Essas mídias podem ser enviadas em alta, média e baixa definição, simultaneamente, tanto para decodificadores específicos na rede quanto para a Internet.

6.3.2 Componentes e Serviços

Os principais componentes e serviços oferecidos pela Arthron são detalhados nesta subseção.

- **Sistema de gerenciamento do usuário (WebService):** É o componente responsável pela gerência e autenticação de usuários e sessões.
- **Articulador (Manager):** Esse componente é responsável pelo controle dos fluxos de mídia e o que será exibido em cada Decodificador e VideoRoom.
- **Codificador (Encoder):** Esse componente é responsável por codificar e enviar um fluxo de mídia, gerado a partir da leitura de um arquivo de mídia existente no computador ou da captura de mídia externa por intermédio, por exemplo, de uma câmera ou placa de captura.
- **Decodificador (Decoder):** Esse componente é responsável por receber e decodificar fluxos de mídia e exibí-las ao espectador.

- **Agente para Videoconferência (VideoRoom):** É um componente com funções de codificador e decodificador, podendo decodificar e exibir vários fluxos de mídia ao mesmo tempo, com a finalidade de prover conferências em alta definição.

6.3.3 Instalação e Configuração

Nesta subseção apresentamos as informações para instalação e configuração da Arthron. A Tabela 02 apresenta os requisitos mínimos do sistema desejáveis para instalação da Arthron.

Tabela 2: Requisitos mínimos do sistema para instalação da Arthron

Componente	Requisito
Sistema Operacional	Ubuntu 11.04 Desktop (32 ou 64 bits)
Processador	Core 2 Duo ou superior
Memória RAM	4 GB ou mais
Disco Rígido	120 GB
Webcam	Compatível com Linux Wireless IP Camera - IP Camera Kit Consist of Wired 12- USB Slot Há a possibilidade de utilizar webcams USB (http://www.linuxtv.org/wiki/index.php/Webcam_Devices)
Componentes Adicionais	Java Runtime Environment (JRE) 1.6 VLC Media Player 1.1.9 Libavcodec-extra-52 (Codec de Áudio e Vídeo) Postgresql (Banco de Dados)

Observações:

- É de responsabilidade do usuário a utilização da Arthron em qualquer distribuição e versão do Linux diferente da recomendada.
- Não aconselhamos o uso de webcams com baixa resolução em cirurgias ou em aplicações que exigem imagens de alta qualidade.
- O computador do usuário deve estar conectado a uma rede;
- A resolução mínima do monitor para o articulador é de 1440x900.

6.3.3.1 Instalação do JRE da Sun/Oracle

Primeiramente deve-se editar o arquivo “sources.list” em /etc/apt/sources.list. O editor utilizado foi o “gedit”, mas pode ser utilizado qualquer editor de texto de sua preferência.

```
$ sudo gedit /etc/apt/sources.list
```

Obs.: O editor utilizado foi o “gedit”, mas pode ser utilizado qualquer editor de texto de sua preferência.

Os comentários das linhas devem ser retirados conforme ilustração da Figura 2 e da Figura 3.

Antes:

```
# deb http://archive.canonical.com/ubuntu natty partner  
# deb-src http://archive.canonical.com/ubuntu natty partner
```

Figura 2: Edição do arquivo sources.list (linhas comentadas)

Depois:

```
deb http://archive.canonical.com/ubuntu natty partner  
deb-src http://archive.canonical.com/ubuntu natty partner
```

Figura 3: Edição do arquivo sources.list (linhas sem comentário)

Salve o arquivo *sources.list* e execute o comando abaixo:

```
$ sudo apt-get update  
$ sudo apt-get install sun-java6-jre
```

6.3.3.2 Instalação de outras dependências necessárias

Dependência da Hauppauge:

```
$ sudo apt-get install ivtv-utils
```

Instalar o VLC:

```
$ sudo apt-get install vlc
```

Codecs de Áudio e Vídeo:

```
$ sudo apt-get install libavcodec-extra-52
```

6.3.3.3 PostgreSQL

Para instalação do sistema gerenciador de banco de dados PostgreSQL deve-se executar a seguinte linha de comando:

```
$ sudo apt-get install postgresql
```

Obs.: O tipo de configuração do servidor de mail deve ser “Sem Configuração”.

O próximo passo é criar o usuário *gtavcs*. Deve-se criar um usuário sem permissão para criação de bancos de dados (-D), sem permissão para adicionar usuários, indica que o usuário será um super-usuário (-s), e pedirá para que o usuário digite uma senha no terminal. Deve-se executar o comando abaixo:

```
$ sudo -u postgres createuser -D -A -P -s
```

Depois, deve-se criar um banco de dados *gtavcs* com o *owner* (-O) sendo o usuário *gtavcs*:

```
$ sudo -u postgres createdb -O gtavcs gtavcs
```

Obs: A senha do WebService deve ser “gtavcs” (sem as aspas).

6.3.3.4 Download da Arthron

Para realizar o download da Arthron execute o comando:

```
$ wget http://h264.lavid.ufpb.br/~erick/Arthron2.5.tar.gz
```

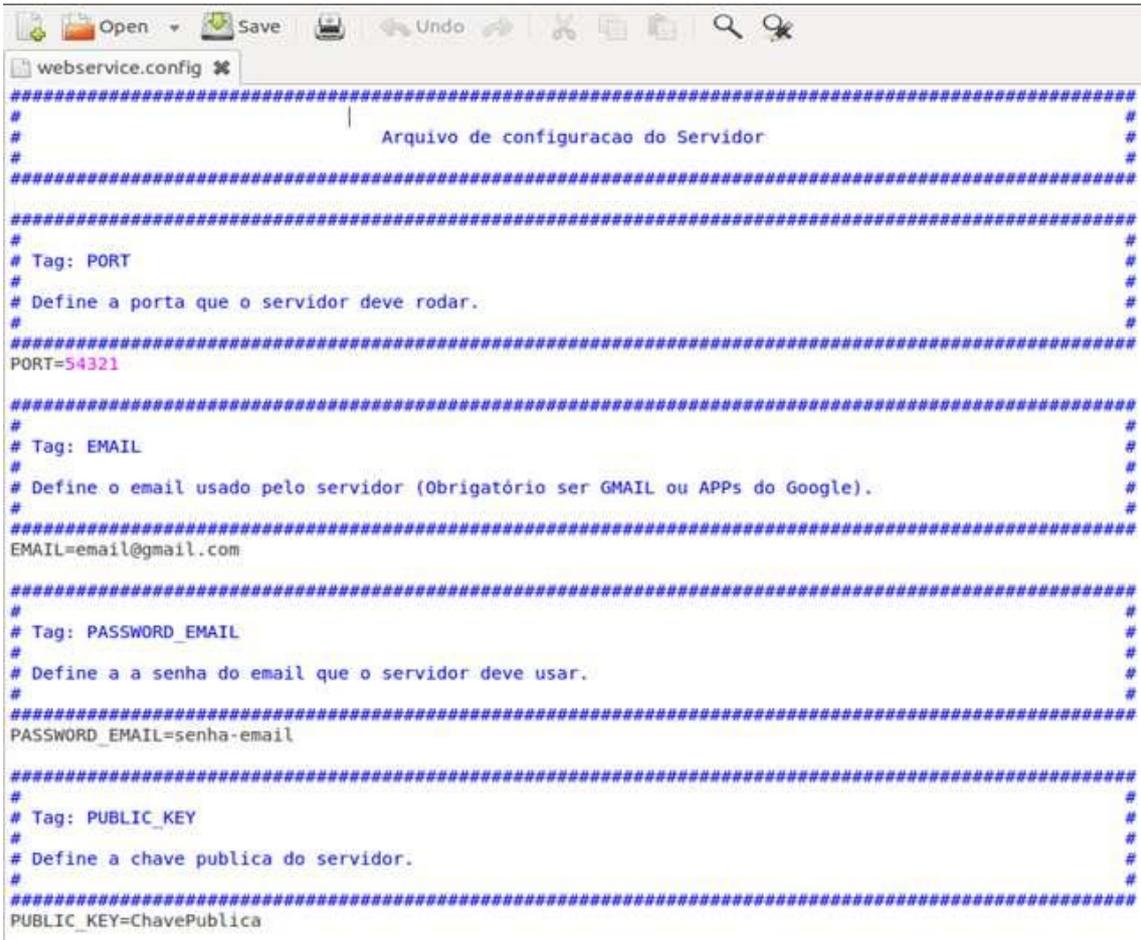
O comando acima irá realizar o download da Arthron em um arquivo compactado que pode ser descompactado com o comando abaixo (a letra *z* deve ser usada porque o arquivo foi compactado com *gzip*; a letra *x* indica que o comando deve extrair o arquivo; a letra *v* exibe os detalhes do procedimento; por fim, a letra *f* especifica qual arquivo será usado nesta atividade):

```
$ tar -zxvf Arthron2.5.tar.gz
```

6.3.3.5 Configuração do módulo WebService

O WebService é o modulo responsável por gerenciar usuários, autenticações e sessões. É responsável pelo armazenamento do perfil do usuário, como nome, email, etc. e histórico das sessões (criador, usuários participantes, data e hora da criação e do encerramento).

É necessário configurá-lo por intermédio do arquivo “WebService.config”, que se encontra no diretório “WebService”. Nesse arquivo é possível alterar a porta que o WebService vai utilizar para a comunicação; o e-mail e senha do e-mail que vai ser utilizado para comunicar os usuários; e a chave RSA, que é composta de três números: chave pública, chave privada e um terceiro número denominado chave N que faz parte das duas chaves. Esses números não podem ser aleatoriamente atribuídos, todos tem que atender as regras do RSA (*fonte: <http://people.csail.mit.edu/rivest/Rsapaper.pdf>*).



```
#####
#
#                               Arquivo de configuracao do Servidor
#
#####
#
# Tag: PORT
# Define a porta que o servidor deve rodar.
#
PORT=54321
#####
#
# Tag: EMAIL
# Define o email usado pelo servidor (Obrigatório ser GMAIL ou APPs do Google).
#
EMAIL=email@gmail.com
#####
#
# Tag: PASSWORD_EMAIL
# Define a a senha do email que o servidor deve usar.
#
PASSWORD_EMAIL=senha-email
#####
#
# Tag: PUBLIC_KEY
# Define a chave publica do servidor.
#
PUBLIC_KEY=ChavePublica
#####
```

Figura 4: Arquivo de configuração do WebService

6.3.3.6 Alertas

A Arthron possui um controle de versão que avisa, na hora da inicialização do componente, se há uma nova versão disponível do software (Figura 5).

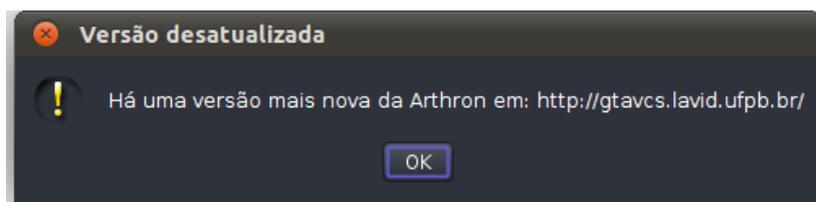


Figura 5: Indicador de nova versão

Outro alerta automático da ferramenta é o controle de versão que não permite que módulos com versões diferentes se conectem (Figura 6).

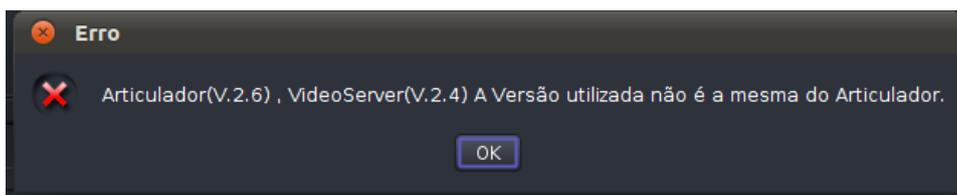


Figura 6: Módulos com versões diferentes tentando se conectar

6.3.4 O Codificador

Este é o componente cuja função é gerar os fluxos de mídia que serão envolvidos na transmissão. O codificador permite a utilização de diversos dispositivos de captura para geração de fluxo e permite, também, a utilização de arquivos existentes no computador. A Figura 7 ilustra o Codificador assim que inicializado.

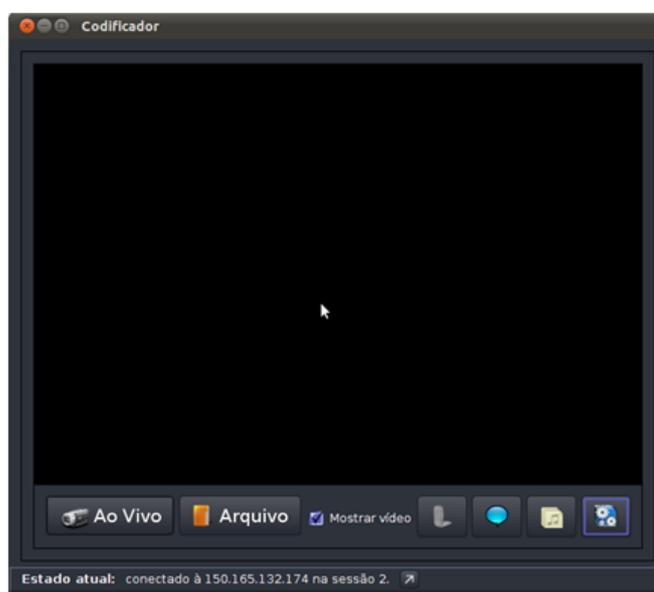


Figura 7: O Codificador

6.3.5 O Decodificador

Este componente tem a função de receber os fluxos de mídia, decodificá-los e exibí-los para o usuário. A Figura 8 ilustra um Decodificador em execução exibindo um fluxo de mídia.

O Decodificador é um componente passivo. Então, ele apenas ficará esperando que as mídias sejam enviadas por ordem do Articulador, não necessitando, portanto, de manipulação mais intensa do usuário. Ele foi feito para ter uma execução rápida e para ser discreto, por isso a sua cor preta (favorável ao corte seco). Para dar um destaque ainda maior à mídia sendo exibida, pode-se “apagar a luz”, clicando no ícone da lâmpada no canto inferior direito da tela, fazendo com que a interface gráfica do Decodificador seja “escondida”.

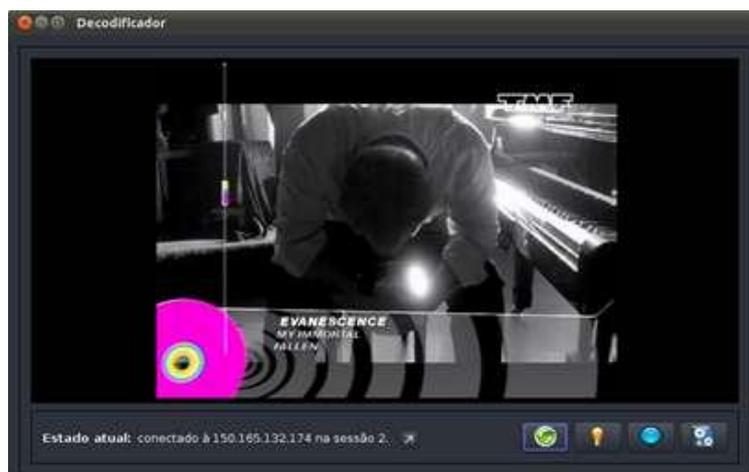


Figura 8: Decodificador exibindo fluxo de mídia

6.3.6 O VideoRoom

Este componente reúne as funções de codificador e decodificador, sendo a melhor alternativa para a realização de videoconferências com a Arthron. A Figura 9 ilustra o VideoRoom assim que conectado ao Articulador.

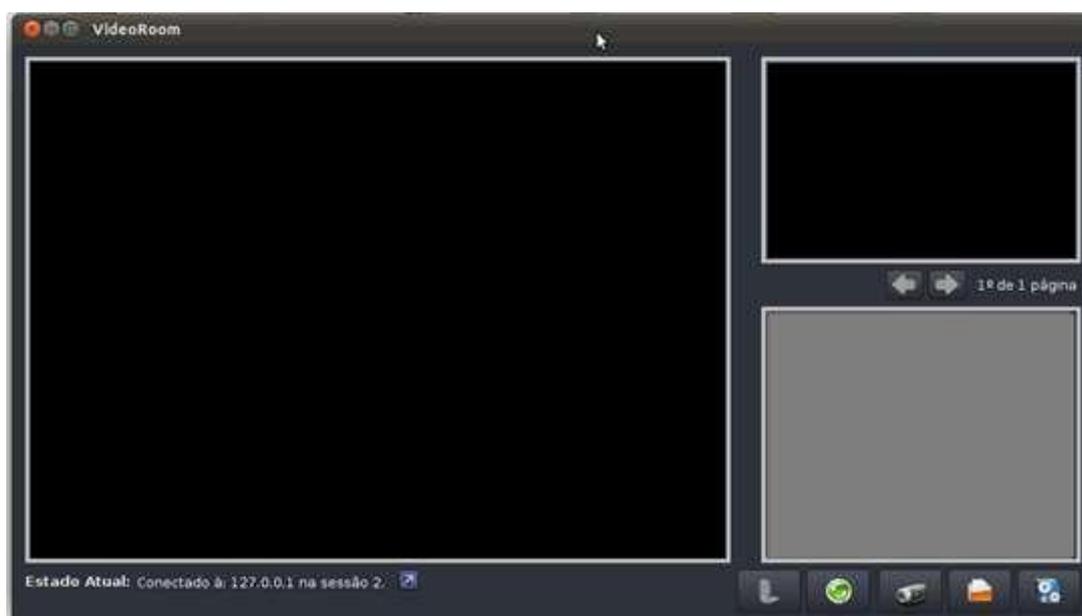


Figura 9: Tela do VideoRoom

A Figura 10 apresenta a tela do VideoRoom após o carregamento de vídeo para transmissão. Na tela maior está localizado o vídeo principal, recebido com a qualidade original. O vídeo em miniatura no canto superior direito trata-se do vídeo capturado localmente. No canto inferior direito temos as miniaturas que correspondem a fluxos secundários, recebidos com qualidade inferior a original. É permitida a troca de fluxo secundário para principal, clicando na miniatura correspondente e arrastando para a tela principal.



Figura 10: VideoRoom exibindo fluxos

6.3.7 O Articulador

Este é o componente cuja função é gerenciar os Codificadores, Decodificadores e VideoRooms nele conectados. A Figura 11 ilustra a janela principal de um Articulador assim que inicializado.

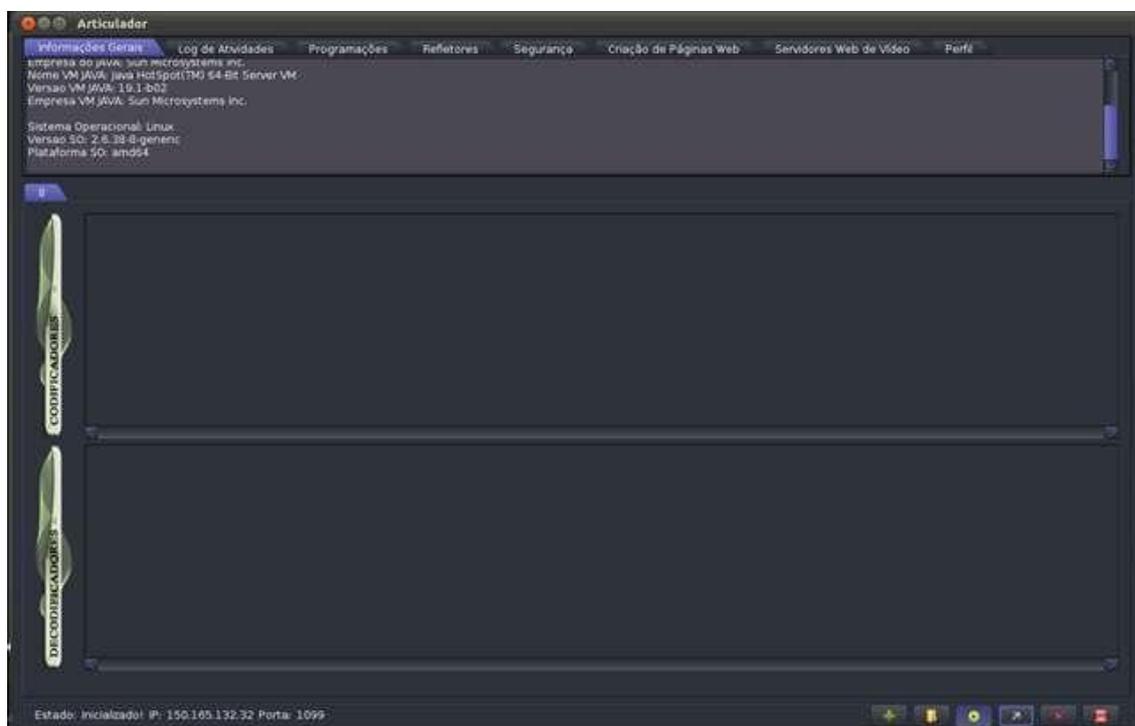


Figura 11: Articulador inicializado

O menu inferior, apresentado na Figura 12, possui ícones para criar nova sessão (1), abrir sessão existente (2), controlar a exibição das miniaturas dos fluxos da sessão (3), exibir informações da sessão (4), fechar sessão (5) e encerrar sessão (6).

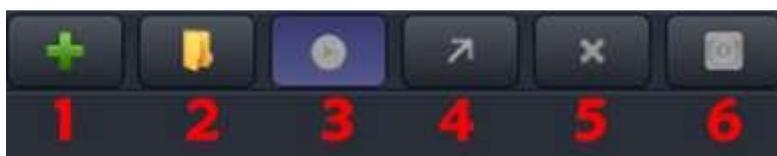


Figura 12: Menu inferior do Articulador

Uma vez inicializado o Articulador estará pronto para receber conexões de todos os componentes e cada um poderá ser gerenciado por intermédio do próprio Articulador. A Figura 13 ilustra o Articulador com um Codificador e um Decodificador conectados à sessão 7.

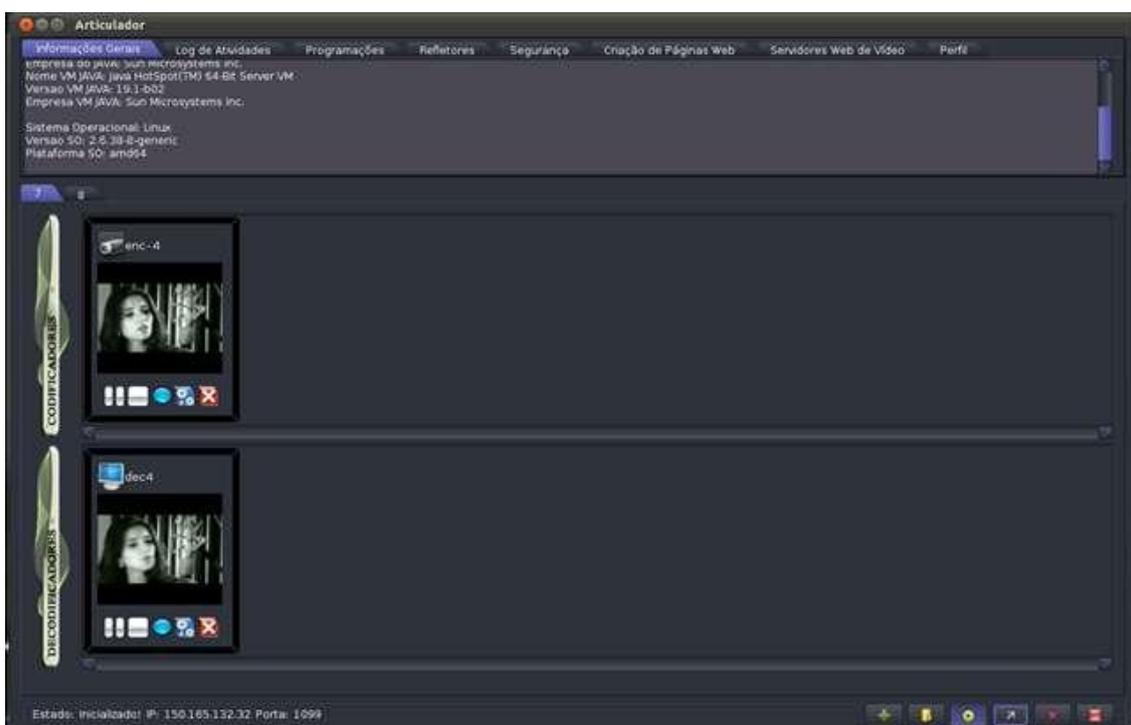


Figura 13: Articulador com codificador e decodificador conectados

6.3.8 Inicialização dos componentes

O primeiro passo para inicializar a Arthron é executar o arquivo `WebService.sh`, localizado na pasta `WebService`. Em seguida deve-se inicializar o Articulador, executando o arquivo `Articulador.sh`, na pasta `Articulador`, para que ele se configure automaticamente.

Obs.: Quando da primeira inicialização, o usuário 'gtavcs' é criado, por padrão, com senha '730985083' (o usuário e a senha devem ser digitados sem as aspas).

A inicialização dos componentes da Arthron segue um mesmo padrão para todos os componentes. Para inicializar corretamente um componente, execute o arquivo `<componente.sh>` localizado em sua respectiva pasta no diretório que você extraiu o programa. A Figura 14 ilustra a janela de autenticação que deverá aparecer. No campo 'Servidor' o usuário deverá digitar o IP do `WebService` responsável pela autenticação.

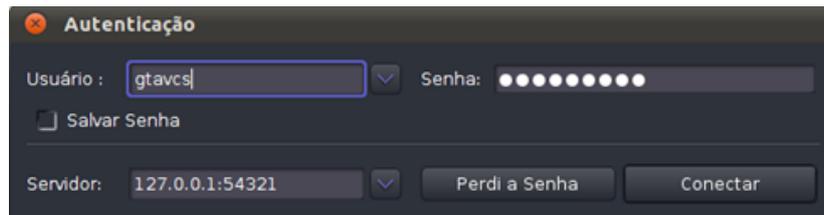


Figura 14: Janela de autenticação

Depois da autenticação, a janela da Figura 15 será aberta. Nela, será possível iniciar o processo de conexão com o Articulador. Para conectar-se a um Articulador, é necessário que o endereço do Articulador esteja cadastrado. No Codificador, por exemplo, para cadastrar um Articulador, deve-se abrir a janela de configuração clicando no botão de configurações destacado na Figura 15.

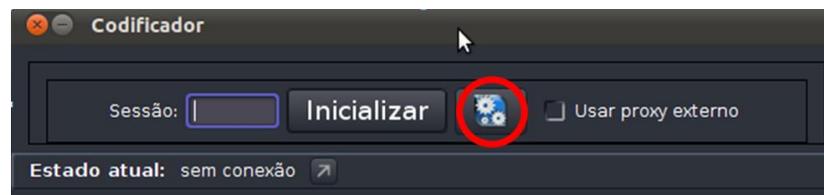


Figura 15: Janela de inicialização do Codificador (botão de configurações em destaque)

Digite o endereço do Articulador no campo **IP/Hostname** e clique em adicionar depois no botão salvar. Agora basta inserir o número da sessão a qual deseja se conectar e clicar em **Iniciar**. A partir de então, o componente poderá ser controlado pelo Articulador.

Obs.: Instruções sobre criação de sessões na seção 6.3.10.

6.3.9 Criação de novos usuários

Para criar um novo usuário deve-se clicar na aba Perfil no *menu* superior do Articulador (as informações exibidas são do usuário que está *logado* no Articulador), em seguida clicar no ícone de adicionar usuário, destacado na Figura 16.



Figura 16: Botão de adicionar usuário

A tela da Figura 17 **Erreur ! Source du renvoi introuvable.** será exibida. Os campos devem ser preenchidos corretamente com as informações do usuário a ser cadastrado. Para gerar uma senha deve-se clicar no botão destacado abaixo. Após preencher todas as informações deve-se clicar no botão salvar. O usuário receberá por e-mail a confirmação do cadastro com sua senha. Para gerar uma senha aleatória, deve-se clicar no botão destacado abaixo.



Figura 17: Janela de adicionar usuário (em destaque, botão para gerar nova senha)

6.3.10 Criação de sessões

Para enviar e receber fluxos deve-se criar uma sessão para abrigar os usuários participantes de transmissão. Para criar uma sessão deve-se clicar no botão destacado na Figura 18, no canto inferior direito da tela. Deve-se escolher se a sessão será moderada ou não.

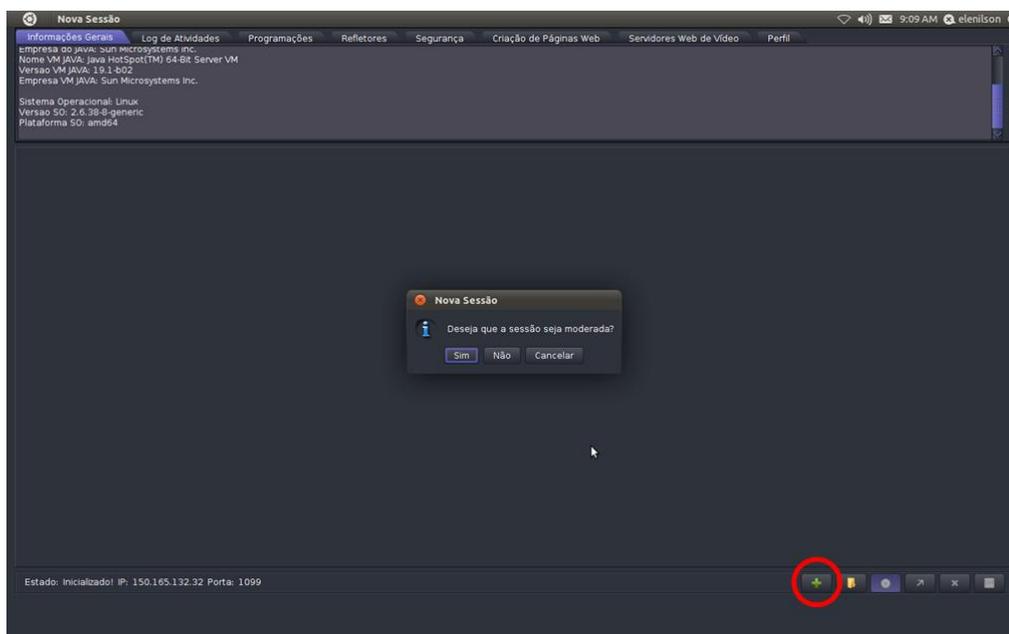


Figura 18: Criando uma sessão

Se a opção escolhida para a sessão for moderada, o gerenciamento dos usuários participantes poderá ser feito clicando no ícone destacado na Figura 19, no canto inferior direito da tela do Articulador. Caso a sessão não seja moderada, o usuário deverá seguir normalmente os passos para transmissão de fluxo e quaisquer usuários poderão requisitar participação na transmissão sem permissão prévia.

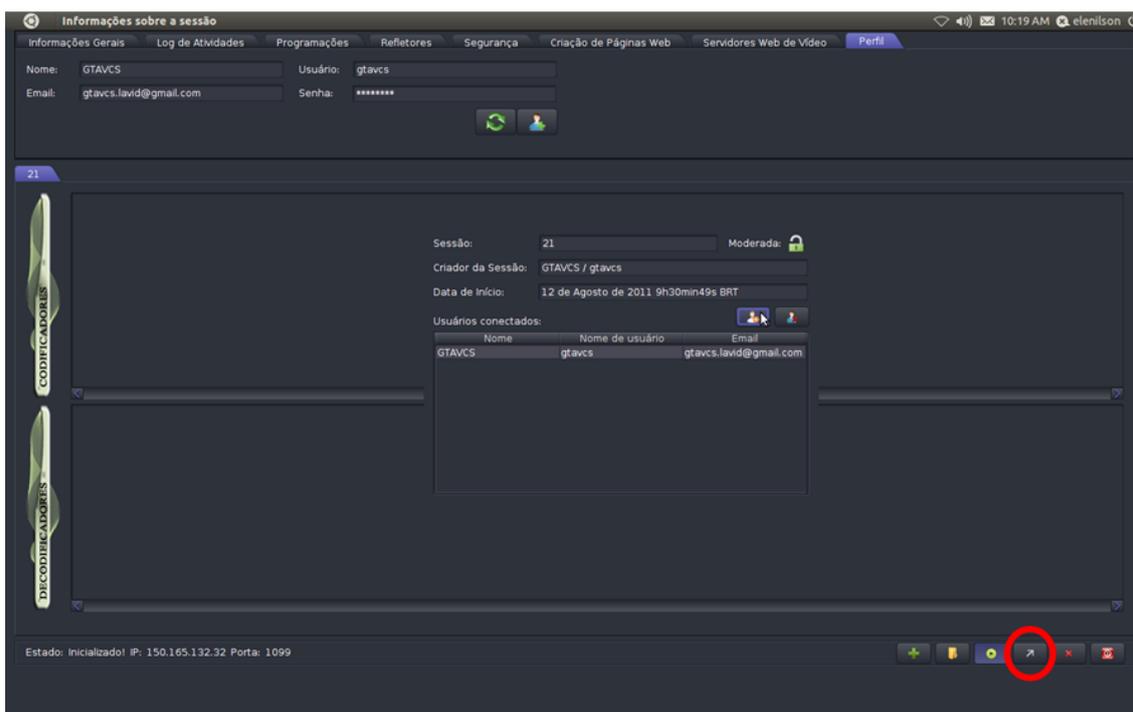


Figura 19: Gerenciamento de usuários da sessão.

6.3.11 Gerando fluxo de mídia com o Codificador ou VideoRoom

Existem duas classes de fluxos de mídia que podem ser gerados, as quais são nomeadas na Arthron de **Ao Vivo** e **Arquivo**. Os fluxos Ao Vivo são capturados em tempo real, como fluxos de webcam e câmera IP, por exemplo. Já os fluxos de Arquivo são gerados a partir de arquivos pré-existentes no computador.

6.3.11.1 Ao Vivo

Para a geração de fluxos de mídia ao vivo, basta clicar no botão **Ao vivo** e será exibida a janela da Figura 20 com as opções de captura. O usuário poderá escolher uma das opções mostradas na figura.



Figura 20: Janela de opções de captura ao vivo

- **Câmera IP:** irá aparecer uma janela solicitando o endereço da câmera conectada remotamente, a porta, o nome do usuário e a senha da autenticação da câmera IP.



Figura 21: Conexão com Câmera IP

- A opção “Stream” permite abrir um fluxo UDP em alguma porta selecionada



Figura 22: Fluxo via stream

- O “Desktop” captura a tela, opção FPS indica quantos números de frames por segundo.



Figura 23: Captura desktop

- A opção “WebCam” só é acionada se estiver conectada alguma WebCam ao computador, escolhe qual a webcam e qual a fonte de áudio que será utilizada.



Figura 24: Captura webcam

- Caso nenhuma das opções listadas seja a desejada, a partir da opção “Outro”, poderá abrir algum outro dispositivo desejado.



Figura 25: Outra opção de captura

- A opção **Avançado** aciona alguma outra característica específica dos dispositivos de captura. Caso essa seja escolhida, deve-se inserir a linha de comando gerada pelo VLC Player (a chamada *MRL – Multimedia Resource Locator*) para abrir o dispositivo em questão.



Figura 26: Opção avançada

- A opção “Firewire” serve para abrir um fluxo em algum dispositivo que esteja conectado em alguma porta *firewire* do computador.

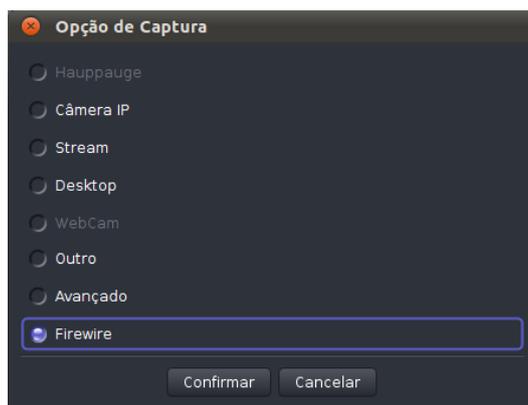


Figura 27: Opção firewire

6.3.11.2 Arquivo

É possível gerar fluxos a partir de arquivos. Para isso, deve-se clicar no botão **Arquivo** e a janela da Figura 28 será exibida para que se possa escolhê-los. Com o arquivo escolhido, deve-se clicar no botão **Abrir** e a respectiva mídia será exibida.



Figura 28: Abrir arquivo no Codificador

6.3.12 Enviando um fluxo

Para enviar o fluxo de vídeo de um Codificador ou VideoRoom para um Decodificador ou VideoRoom, basta clicar no nome do componente na miniatura correspondente no Articulador, manter o botão do mouse pressionado e arrastar o ponteiro até o nome do componente da miniatura correspondente. Então surgirá a tela da Figura 29. Selecione a opção Troca Imediata e clique em **Confirmar**. O Decodificador ou VideoRoom passará a exibir o que o Codificador ou VideoRoom estiver enviando.

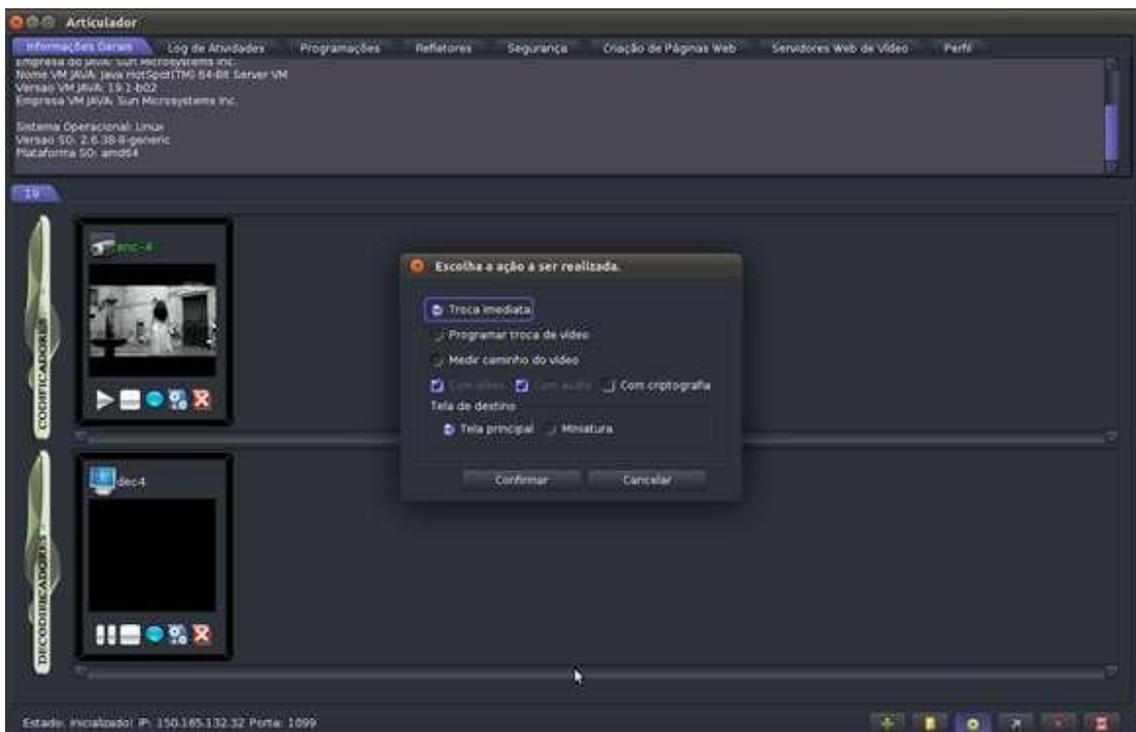


Figura 29: Troca de fluxo entre Decodificador e Codificador

Com a Arthron também é possível enviar um fluxo em tempo programado. O procedimento é semelhante ao envio normal, onde a única diferença é que a opção a ser escolhida é **Programar troca de vídeo** ao invés de **Troca imediata**, como é demonstrado na Figura 30.



Figura 30: Opção para programar a troca de vídeo

Após clicar em **Confirmar**, surgirá a janela da Figura 31, onde deverá ser informado o tempo que o Codificador ou VideoRoom deverá esperar para começar a enviar o fluxo para o Decodificador ou VideoRoom. Depois de informado o tempo, pressione o botão **Confirmar**.

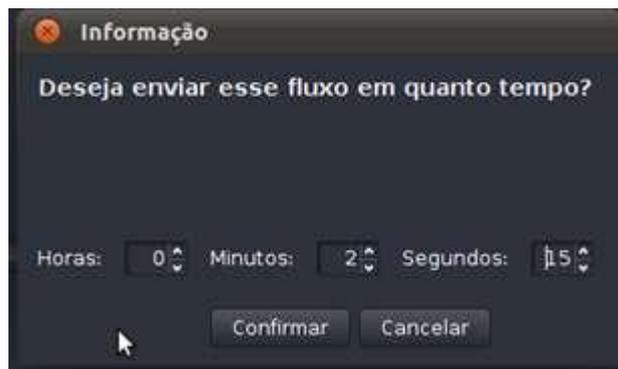


Figura 31: Temporizador de envio de fluxo

6.3.13 Visão do VideoRoom no Articulador

A Figura 32 ilustra o VideoRoom dividido em duas partes no Articulador. No painel **Codificadores**, a miniatura do VideoRoom exibe o que está sendo capturado pelo VideoRoom, ou seja, o codificador. No painel **Decodificadores**, a miniatura do VideoRoom exibe todos os fluxos recebidos pelo VideoRoom. É possível visualizar todos os fluxos que estão sendo recebidos, um por vez, apertando no botão de trocar, indicado em destaque na Figura 32.

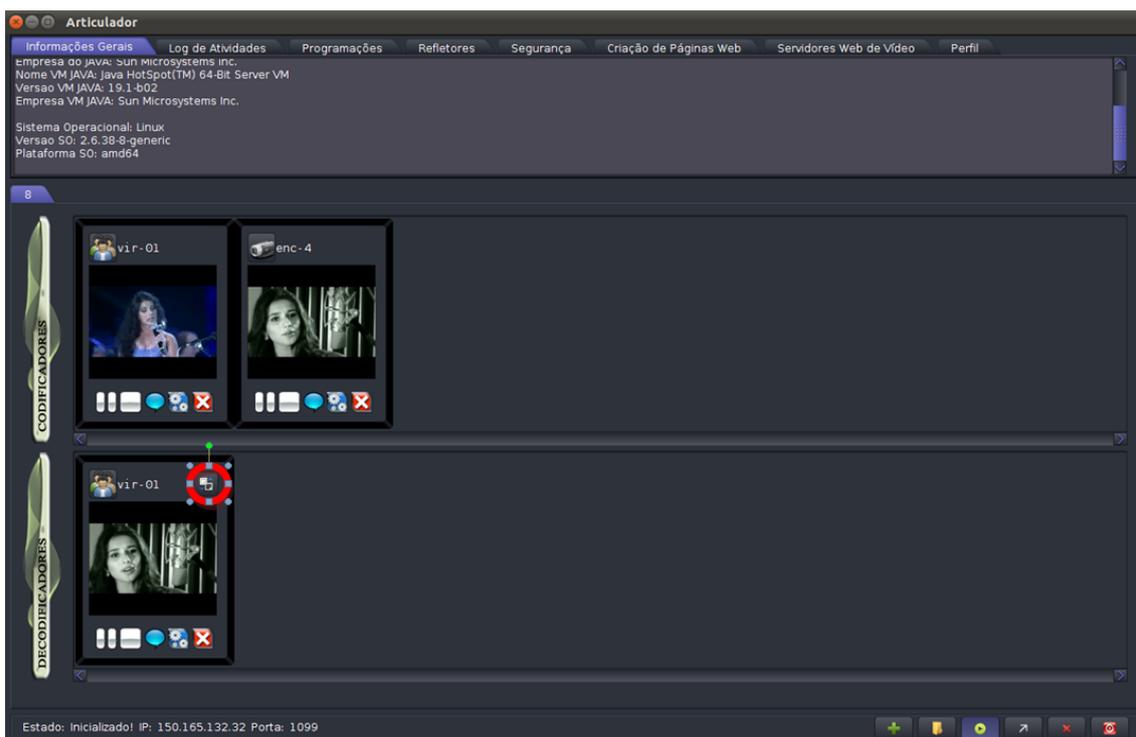


Figura 32: Representação do VideoRoom no Articulador (em destaque, botão para chaveamento entre fluxos recebidos pelo VideoRoom)

6.3.14 Mudando as configurações de transcodificação de mídia

É possível, nas configurações, estabelecer tipos de codificação pré-definidos de áudio e vídeo, bem como as suas respectivas taxas de transmissão. Isso pode ser visto na Figura 33.

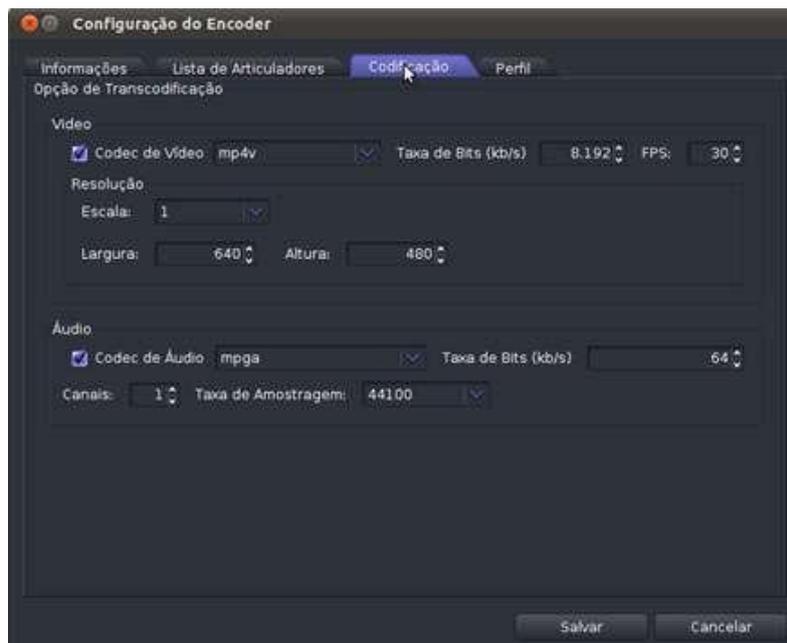


Figura 33: Configuração da transcodificação

6.3.15 Abas do Articulador

O Articulador possui diversas abas que ajudam no gerenciamento da transmissão. As principais são descritas a seguir.

Obs.: A aba Refletores não será descrita neste documento e a aba Criação de Páginas Web está contida na próxima seção.

6.3.15.1 Informações Gerais

Descreve as informações gerais, tais como sistema operacional, versão do SO, plataforma do SO e outras informações a respeito da Máquina Virtual Java.

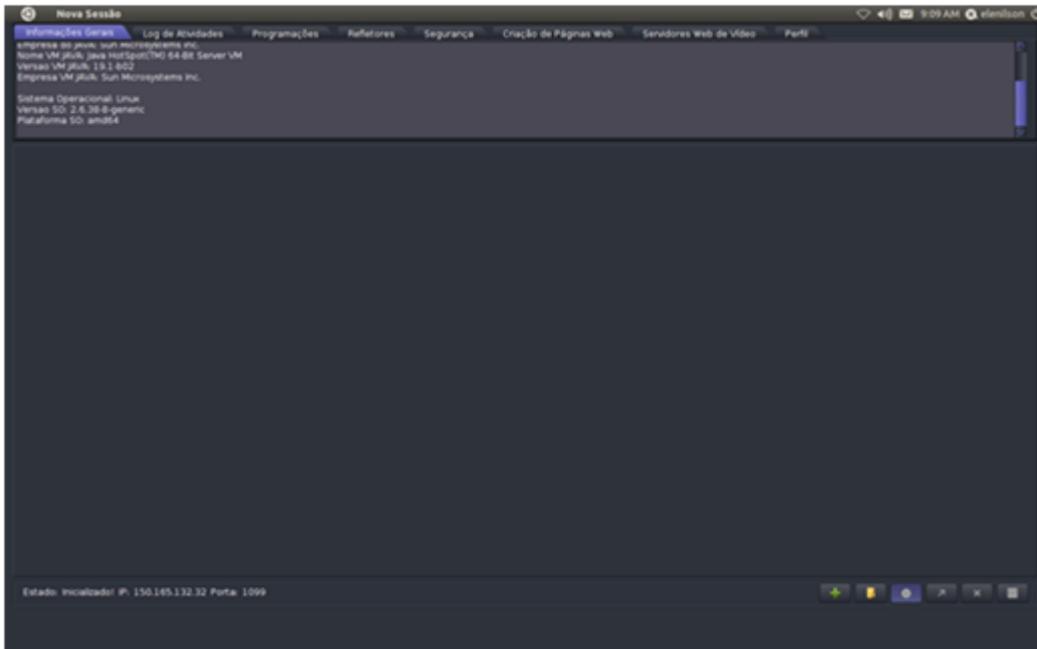


Figura 34: Articulator – aba Informações Gerais

6.3.15.2 Log de Atividades

Descreve as principais atividades realizadas na Arthron, tais como conexão de Codificadores, Decodificadores e Refletores, programação de trocas de vídeos, etc. Essas informações são salvas em um arquivo na pasta *log* que está dentro da pasta do Articulator. Esse arquivo obedece ao formato *log_<DD-MM-AAAA_hh_mm_ss.txt>*, onde *DD* é o dia, *MM* é o mês, *AAAA* é o ano, *hh* é a hora, *mm* é os minutos e *ss* é os segundos em relação a data de criação do log.

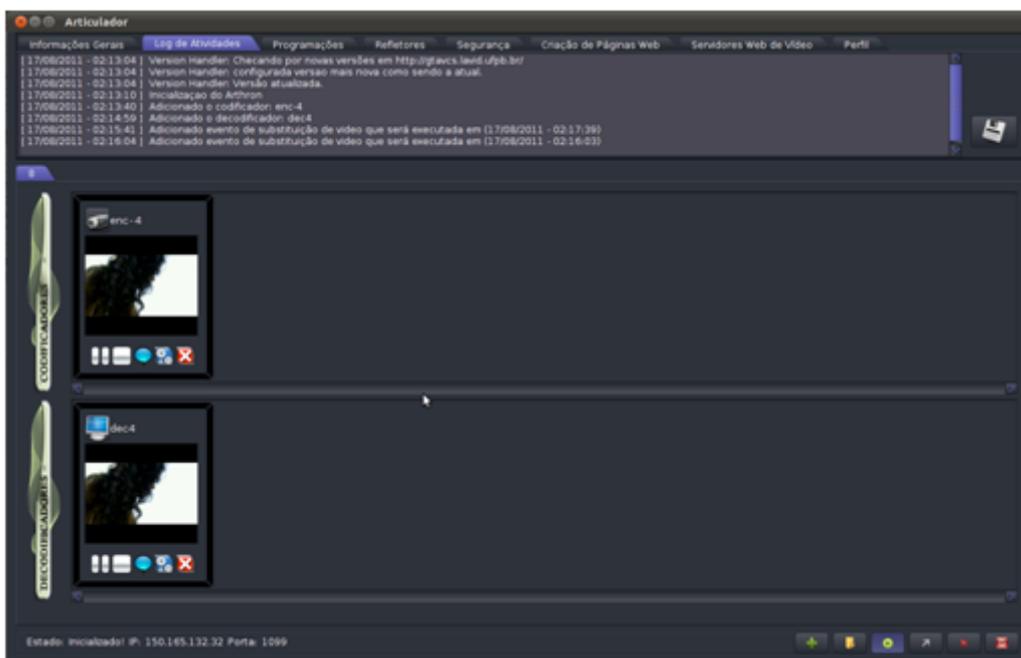


Figura 35: Articulator – aba Log de Atividades

6.3.15.3 Programações

Ilustra graficamente os eventos temporais que foram cadastrados (Figura 36). Também permite a visualização em uma tabela (Figura 37), clicando-se no botão que se encontra a direita nessa aba. Os eventos temporais possíveis são atribuição de vídeo de um Codificador a um Decodificador, troca de vídeo entre dois Decodificadores e tempo de início de uma animação em um Decodificador.

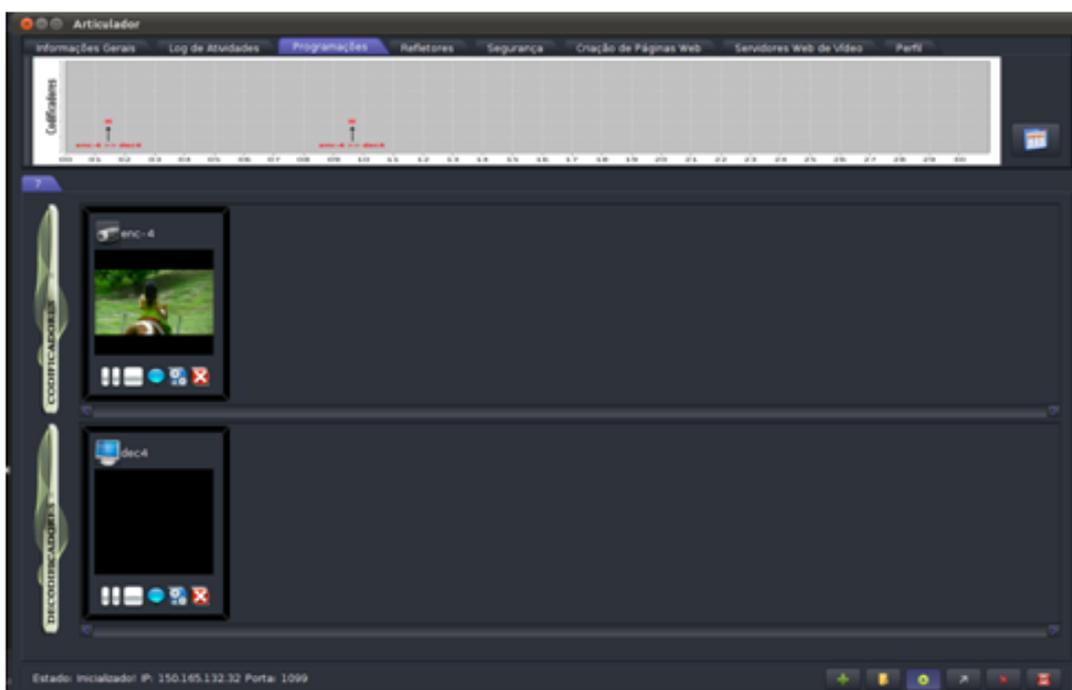


Figura 36: Articulador – aba Programações

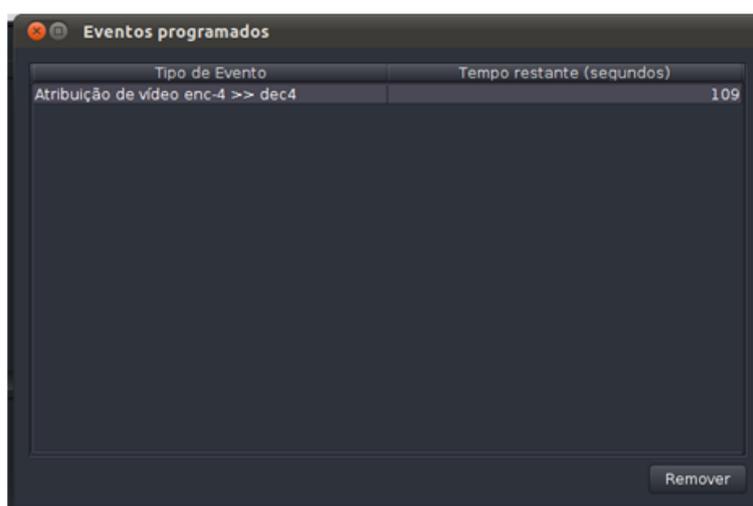


Figura 37: Eventos Programados

6.3.15.4 Segurança

Permite evitar a conexão de Decodificadores, Codificadores ou Refletores que se originem de um determinado IP.

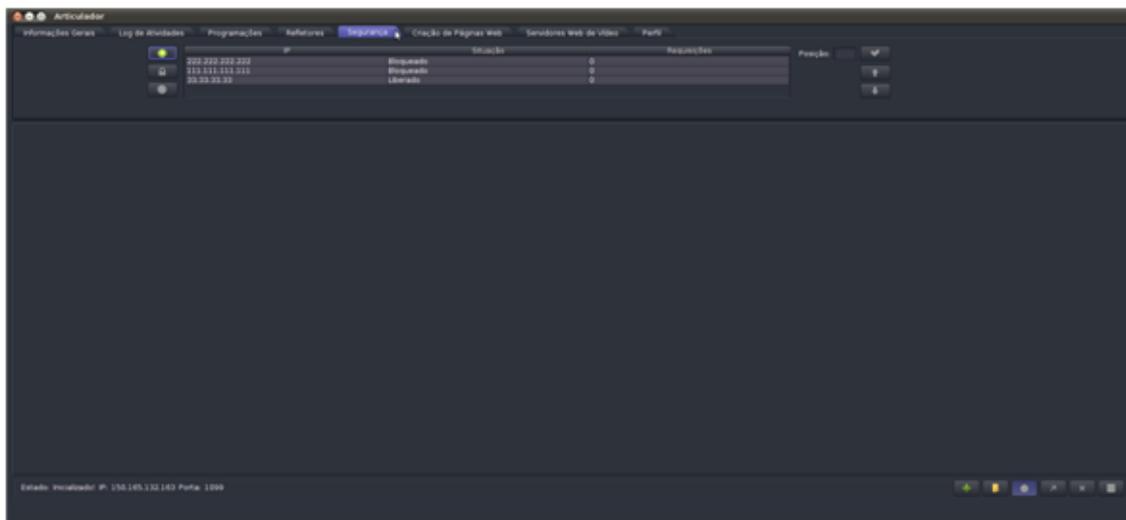


Figura 38: Articulator – aba Segurança

6.3.15.5 Perfil

Exibe os dados do usuário que está *logado* no articulador.

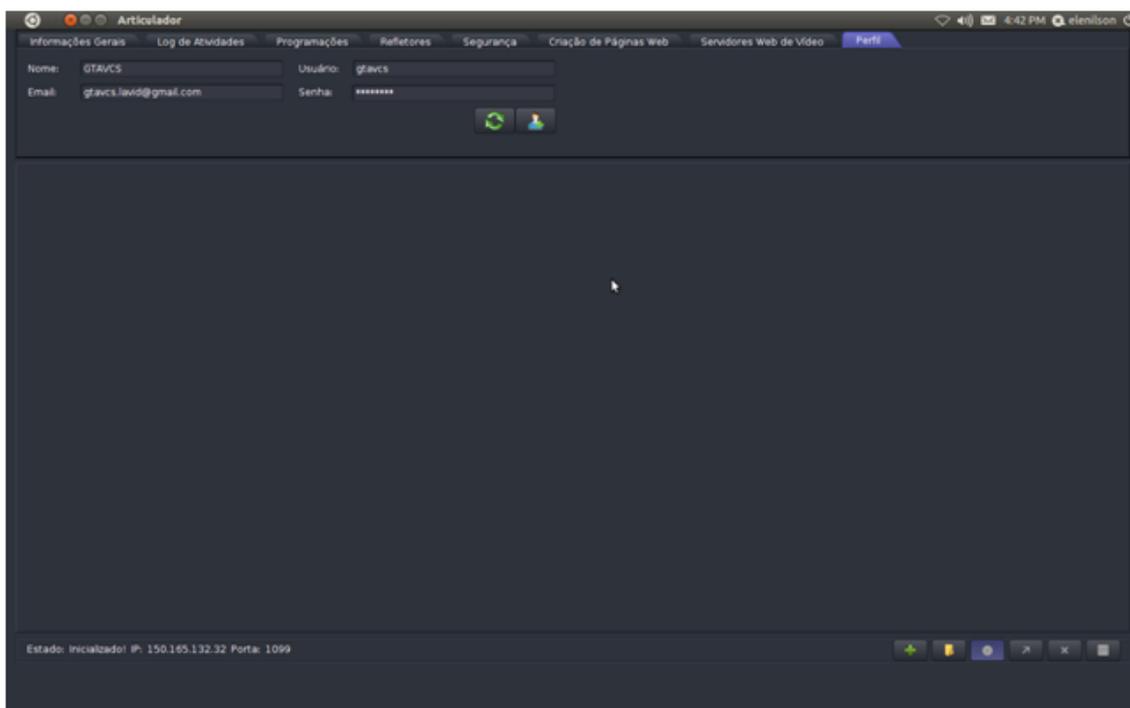


Figura 39: Articulator – aba Perfil

6.3.16 Publicando vídeos na Web utilizando o VideoServer

A função de um servidor de vídeo é transmitir o vídeo para clientes web. Para criar um servidor de vídeo é necessário inicializar o componente de servidor de vídeo (VideoServer). Irá aparecerá a janela da Figura 40. Preencha os campos de identificação, IP, porta RMI e porta Transmissão.



Figura 40: VideoServer

Ao clicar no botão **Iniciar**, o servidor de vídeo deve ficar esperando conexão. Na aba “Servidores Web de Vídeo” do Articulador ou pelo caminho descrito anteriormente em **Publicação na Web**, o usuário deve clicar no botão **adicionar**, abrirá uma janela com as opções de servidor de vídeo (Figura 41). O usuário deve preencher com o IP do servidor de vídeo e a porta RMI, então a conexão com o servidor de vídeo será estabelecida.

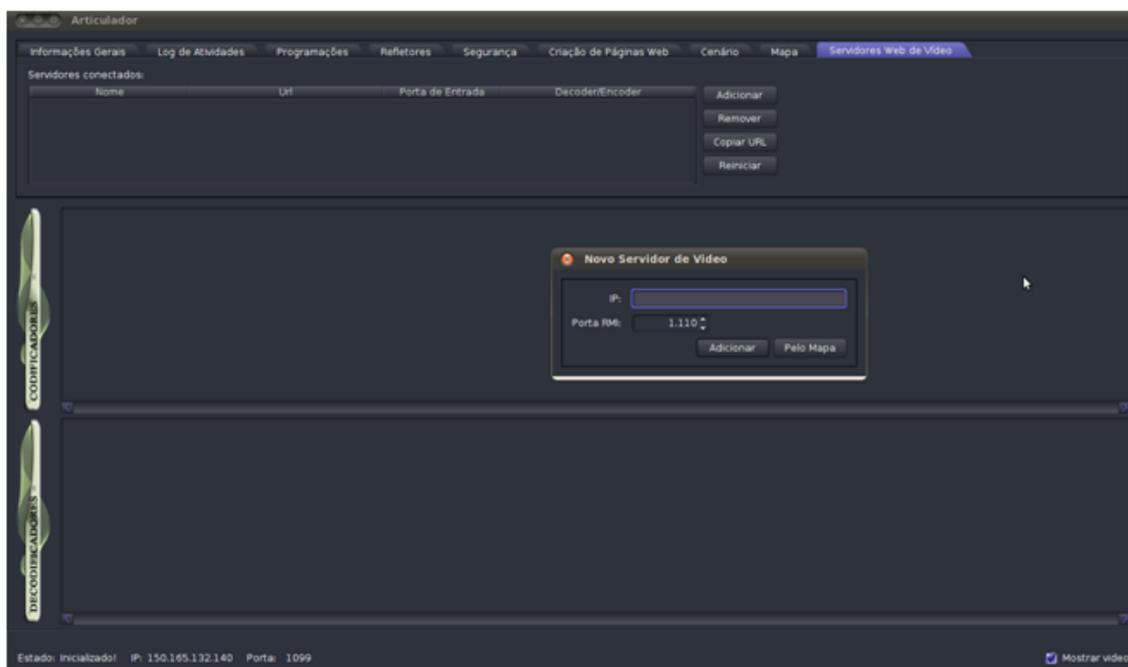


Figura 41: Articulador: adicionando um Servidor Web de Vídeo

6.3.16.1 Criação de página Web

O usuário poderá criar uma página web com os vídeos que estão sendo exibidos nos decodificadores selecionados. Depois de publicar na web a mídia desejada, abra a aba “Criação de Páginas Web” e escolha o fluxo que está sendo publicado.

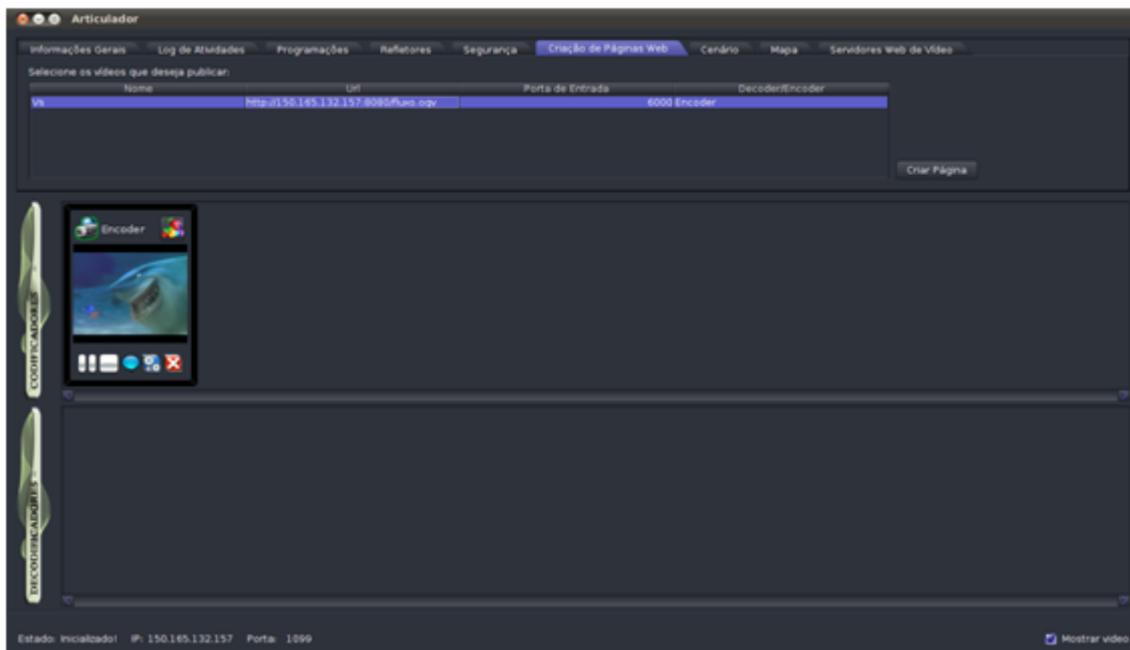


Figura 42: Articulator – aba Criação de Páginas Web

Ao clicar em **Criar Página** irá aparecer a tela exibida na Figura 43.



Figura 43: Criação de página Web

- Template: Selecionar a opção com o template escolhido.
- Textos: Escrever o título da página, o rodapé e uma descrição para ela.
- Vídeos: Escolher o tamanho dos players.
- Logo: Adicionar a imagem para a Logo e o tamanho dessa imagem.
- Google Analytics: Ao acessar a página www.google.com/analytics e seguir as instruções, um código será gerado, copie esse código e cole no campo.
- HTML Personalizado: Escrever um código HTML que será exibido abaixo do player.
- Áudio em Background: Insira um Link de um áudio no formato **ogg**.

Após clicar em **Criar**, escolha o local que será salva a pasta com os componentes necessários de uma página na web. Essa página será composta por um *menu* lateral com os fluxos que foram previamente selecionados. Clique na mídia que deseja assistir, o

vídeo deve ser exibido no centro da página (Figura 44). Veja abaixo o exemplo de uma página da web.



Figura 44: Página Web com template verde

6.3.17 Bate-papo

O bate-papo é um sistema que provê a comunicação entre os controladores do Articulador, Codificador e Decodificador.

O ícone de bate-papo é representado por um balão azul. Ao clicar nele irá aparecer uma janela de conversação. Obs.: Caso seja no Articulador a janela de conversação aparecerá embutida na própria janela de transmissão do Codificador ou Decodificador. Para desmembrar clique no ícone indicado na Figura 45.



Figura 45: Botão para iniciar desmembrar o bate-papo do Articulador

Caso o usuário seja o controlador do Articulador, se clicar no ícone presente na janela de transmissão do Codificador, irá se comunicar com o controlador do Codificador e se clicar no ícone presente na janela de transmissão do Decodificador, irá se comunicar com o controlador do Decodificador.

Caso o usuário seja o controlador do Codificador ou Decodificador, ao clicar no ícone de bate-papo, irá se comunicar com o controlador do Articulador.

Quando o Articulador recebe uma mensagem, a janela de exibição ficará na cor laranja, sinalizando que tal componente enviou uma mensagem via Bate-Papo (Figura 46).



Figura 46: Articulador recebeu mensagem de um componente

Quando o Codificador ou Decodificador recebe uma mensagem, aparecerá a janela de conversação com a nova mensagem (Figura 47).

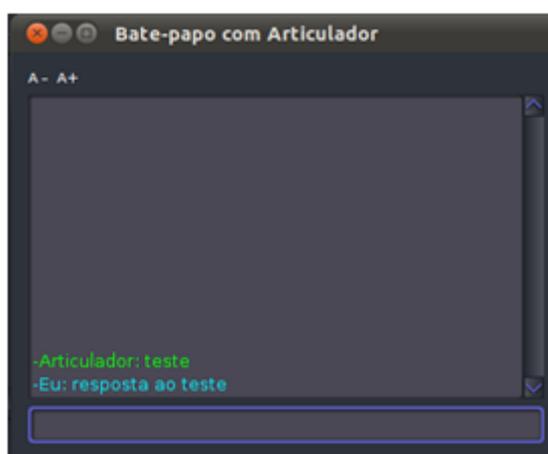


Figura 47: Bate-papo com o Articulador

6.4 Conclusão

O uso de ferramentas especializadas para o gerenciamento de fluxos envolvidos em eventos artísticos, esportivos ou em experimentos como telemedicina ou teleaula diminui a complexidade operacional e aumentam as possibilidades de transmissão. A Arthron é, então, uma ferramenta que visa contemplar esses cenários como uma solução integrada para gerenciar fluxos midiáticos distribuídos em tempo real. A ferramenta provê o suporte à articulação entre diferentes fontes midiáticas, facilitando o modo de interação entre esses agentes e os agentes humanos, ou seja, com apenas computadores, câmeras e/ou projetores é possível qualquer usuários fazer a sua própria transmissão de uma forma simples e barata.

6.5 Agradecimentos

Agradecemos à RNP (Rede Nacional de Ensino e Pesquisa) pelo apoio financeiro para realização dos GTs, especialmente, GTMDA e GTAVCS. Agradecemos ao GP Poética pela contribuição e concepção artística essencial para realização do E-Pormundos Afeto.

Por fim, agradecemos ao Hospital Universitário Lauro Wanderley pela parceria fundamental para realização deste projeto.

Referências

- Austerberry, D. The technology of video and audio streaming. 2nd ed. Elsevier. Burlington, USA. 2005.
- Dukes, J.; Jones, J. Using Dynamic Replication to Manage Service Availability in a Multimedia Server Cluster in Interactive Multimedia and Next Generation Networks. Springer Link, Grenoble, France. 2004.
- Melo, E.; Pinto, A.; Silva, J.; et al.. Arthron 1.0: Uma ferramenta para transmissão e gerenciamento remoto de fluxos de mídia. 2010.
- Morris, T. Multimedia Systems: delivering, generating, and interacting with multimedia. Springer-Verlag London Limited 2000.
- Santana, I. A imagem do corpo através das metáforas (ocultas) na dançatecnologia. Belo Horizonte - Minas Gerais - Brasil. : s.n., 2003. INTERCOM – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação. p. 17.