

MyPersonal-EPG: Um EPG Personalizável e com Suporte à Recomendações¹

Pedro P.C.Maia

DIMAp/UFRN

Campus Universitário – Natal - RN

+55 84 3215 3814

petro1987@yahoo.com.br

Jair Leite

DIMAp/UFRN

Campus Universitário – Natal - RN

+55 84 3215 3814

jair@ufrnet.br

Thais Batista

DIMAp/UFRN

Campus Universitário – Natal - RN

+55 84 3215 3814

thais@ufrnet.br

ABSTRACT

This paper presents the architecture and implementation of MyPersonal-EPG, an Electronic Programming Guide (EPG) that supports customization and recommendation. It is based on Ginga middleware and it meets the following requirements: (i) creation of user's personal programming guide using data sent by TV programs providers; (ii) automatic channel tuning that runs whenever a selected program is going to start; (iii) definition and selection of programs categories that can be user in recommendations; (iv) synchronous and asynchronous recommendations based on selected program categories; (v) personal configuration of several features; (vi) login accounts that associates personal configurations and preferences to each user.

RESUMO

Neste artigo, apresentamos a arquitetura e implementação do MyPersonal-EPG, um EPG personalizável e com suporte a recomendações, construído utilizando o middleware Ginga, que atende aos seguintes requisitos: (i) permitir aos usuários a montagem da sua própria grade de programação, com base nas grades de programação de diversas emissoras; (ii) oferecer um mecanismo para sintonizar os devidos canais no momento em que os programas selecionados pelo usuário estiverem prestes a começar; (iii) permitir que os usuários selecionem categorias de programas desejadas; (iv) oferecer recomendações de programas, de forma síncrona e assíncrona, com base nas categorias selecionadas previamente pelos usuários; (v) permitir que os usuários modifiquem as opções de configuração; (vi) possibilitar a criação de diversas contas de usuários, a fim de permitir que cada usuário possa armazenar todas as suas informações de interesse.

Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems] – *Video (e.g., tape, disk, DVI)*

General Terms

Design, Human Factors.

Keywords

TV Digital, Guia Eletrônico de Programação, Personalização, Recomendações, SBTVD.

1. INTRODUÇÃO

Uma extensa variedade de programas e serviços vem sendo disponibilizados através da TV Digital. A grande quantidade de conteúdos digitais disponíveis motiva a criação de aplicações que sejam capazes de coletar informações sobre os programas e serviços e apresentá-las de maneira organizada e, se possível, adaptável às necessidades dos telespectadores (daqui por diante chamados de usuários).

As informações sobre programas e serviços são enviadas, por difusão ou sob demanda, multiplexados com outros tipos de dados como, por exemplo, vídeo e áudio. Tipicamente, existe um formato padronizado para os metadados. Os principais sistemas de TV Digital definem um conjunto de tabelas que concentram informações sobre os programas e serviços disponíveis. No sistema Brasileiro de TV Digital (SBTVD), norma ABNT NBR 15603-1 [1], tais tabelas são conhecidas como SI (*Service Information*). Uma dessas tabelas é a *Event Information Table* (EIT), responsável por carregar metadados dos programas que formam a programação. Cada evento consiste em um identificador único, um tempo de início, uma duração e um conjunto de descritores para informações adicionais. Exemplos comuns de informações adicionais são a idade mínima para os telespectadores e o gênero e o subgênero do evento. Os metadados são transmitidos em fluxos de transporte *MPEG2-System* ISO/IEC 13818-1 [9] e seguem o formato padronizado em [2].

Na literatura, aplicações que apresentam informações aos usuários sobre os programas e serviços são divididas em dois tipos: (i) *Guias Eletrônicos de Programação* (EPGs) [11], os quais simplesmente compilam e exibem informações obtidas por difusão e (ii) *Guias Eletrônicos de Serviço* (ESGs) [7], os quais, além de desempenhar as tarefas dos EPGs, também incluem informações de conteúdo que podem ser adquiridas sob demanda, tais como: dados sobre os integrantes da banda, em um show de rock, ou ainda a classificação do campeonato, em um jogo de futebol.

¹ MyPersonal-EPG: A customizable EPG with support to recommendations

Em [10], os autores distinguem uma aplicação EPG/ESG, chamada de *Aplicação de Guia Eletrônico* (EGA), e o resultado de sua execução, a informação efetivamente apresentada pelo guia, chamada de *Informação do Guia Eletrônico* (EGI). Eles argumentam que, em geral, as EGAs apresentam os metadados de maneira adaptada, de acordo com os perfis dos usuários. Além disso, as EGAs comuns também permitem a adaptação do leiaute usado na apresentação dos dados. No entanto, dificilmente a EGA em si é adaptável. Algumas vezes é possível que ela sofra atualizações esporádicas, mas não que sejam adaptadas *on-the-fly*. Em resumo, normalmente as EGIs são adaptáveis, enquanto as EGAs não são. Para resolver esse problema, eles propõem uma arquitetura de EGA modular, que inclui um metasserviço responsável por permitir atualizações dinâmicas.

Devido ao grande número de canais e programas de diferentes gêneros, utilizar uma EPG pode ser uma tarefa enfadonha e decepcionante. O usuário precisa navegar usando o controle remoto por dezenas de canais para visualizar os programas em um determinado horário e data. Para isto, o EPG precisa oferecer mecanismos facilitadores para escolha dos programas e para que os usuários montem a própria grade de programação, incluindo programas de diversas emissoras. Diversas técnicas podem ser empregadas para seleção de programas. Uma delas é a busca de programas, por nome do programa, gênero, dia e horário, etc [15]. Outra técnica possível é a de recomendação de programas [13,14]. Nesta técnica, o EPG usa algumas informações do usuário para recomendar os programas de sua preferência. É interessante que a recomendação seja dada tanto no momento em que o usuário está interagindo com a aplicação, quanto posteriormente, quando novos programas de interesse do usuário sejam disponibilizados nas grades de programação das emissoras.

Nesse artigo propomos o *MyPersonal-EPG*, uma EGA que fornece: (i) recomendações de programas aos usuários, de forma síncrona e assíncrona; (ii) montagem da grade de programação personalizada. Essa grade personalizada exibe as informações referentes aos programas e serviços existentes a partir de escolhas dos usuários ou a partir de recomendações do sistema. De posse das informações da grade, o *MyPersonal-EPG* pode sintonizar automaticamente os canais referentes aos programas selecionados no momento em que eles estiverem sendo exibidos, sem que o usuário tenha que acionar o controle remoto.

A recomendação de programas e serviços no *MyPersonal-EPG* é feita com base em preferências submetidas pelos usuários. Tais preferências são expressas na forma de gênero e subgênero dos programas que os usuários desejam assistir. Os usuários podem, a qualquer momento, inserir e remover novas categorias (cada categoria é formada por um gênero e um subgênero) em seu quadro de categorias selecionadas. O *MyPersonal-EPG* é responsável por fazer sugestões de programas que se enquadrem nas categorias selecionadas. Essas sugestões são feitas tanto sincronamente quanto assincronamente. O mecanismo assíncrono facilita consideravelmente a vida do usuário já que permite a ele simplesmente cadastrar as categorias em que tem interesse e deixar que o *MyPersonal-EPG* atualize a lista de programas recomendados sempre que um novo programa disponibilizado enquadre-se em tais categorias. O *MyPersonal-EPG* permite também mover um programa que está na lista de programas recomendados para a grade de programas selecionados.

O *MyPersonal-EPG* também oferece funcionalidades adicionais que permite ajustes de acordo com as necessidades dos usuários, tais como: (i) se o aparelho de televisão deve ou não ser ligado automaticamente caso se detecte que um dos programas selecionados está prestes a começar; (ii) se os programas selecionados devem ser gravados ou não e (iii) se a troca de canais deve ser automática ou não, quando o programa seguinte na lista de programas selecionados esteja prestes a começar.

O *MyPersonal-EPG* estende a arquitetura apresentada por [10]. A principal motivação em empregar esta arquitetura modular, pela adição de dois novos componentes: *Scheduler* e *Recommender*, é permitir que os componentes que compõem a aplicação possam ser adaptados sem que as mudanças em um componente afetem os demais. Além disso, tais mudanças devem poder ser feitas em tempo de execução para que a aplicação possa estar sempre em funcionamento, de forma que o usuário não precise aguardar por um processo de atualização demorado estando privado do uso da aplicação.

Em termos de implementação, o *MyPersonal-EPG* usa como infra-estrutura subjacente o middleware *Ginga* e, portanto, foi desenvolvido usando a linguagem declarativa NCL (*Nested Context Language*) [3], juntamente com a linguagem de script associada, Lua [8]. O *Ginga* contempla o uso conjunto das duas linguagens e provê: suporte à execução de aplicações NCL e módulos que podem ser usados no desenvolvimento dos componentes em Lua (como os módulos *event* e *canvas*, particularmente importantes no contexto deste trabalho).

Neste trabalho, apresentamos o design da interface de usuário (IU) do *MyPersonal-EPG*, a extensão da arquitetura e implementação do componentes adicionais necessários. O design da IU foi elaborado levando-se em consideração as diretrizes apontadas em [4]. As diretrizes dão sugestões em relação a tamanhos de tela empregados, definição de fontes adequadas para exibir textos na TV, redimensionamento de imagens para exibição adequada na TV Digital, uso de mecanismos de navegação e controle adequados e uso de teclas adequadas do controle remoto. Tais diretrizes foram definidas em [5], a partir da experiência de vários anos em desenvolvimentos de aplicações para TV Digital.

Este artigo está organizado da seguinte forma. A seção 2 apresenta os conceitos básicos necessários ao entendimento desse trabalho. A seção 3 apresenta a arquitetura do *MyPersonal-EPG*. A seção 4 discute os aspectos de implementação. A seção 5 comenta os trabalhos relacionados e a seção 6 traz as conclusões.

2. CONCEITOS BÁSICOS

Nesta seção abordaremos o conceito de recomendações (seção 2.1) e apresentaremos a arquitetura de referência proposta por [10] (seção 2.2), que serviu de base para a arquitetura e do *MyPersonal-EPG*.

2.1 Recomendações em Sistemas Computacionais

O grande volume de informações e opções disponibilizadas para as pessoas traz benefícios, mas, no entanto, traz também dificuldades para se encontrar informações realmente úteis e fazer boas escolhas. Essa situação é conhecida na literatura como problema de *sobrecarga de informações* [12]. Para solucionar esse problema, os sistemas de recomendações têm

sido amplamente empregados. Eles são responsáveis por recomendar itens (filmes, músicas, páginas web, etc.) de interesse dos usuários. Esses sistemas têm sido muito usados em aplicações comerciais e sites de *e-commerce*. Alguns exemplos podem ser citados: o *site* de *e-commerce amazon.com* recomenda vários tipos de produtos (livros, música, dispositivos eletrônicos, etc.) aos seus clientes; o *site last.fm* recomenda músicas; a aplicação *TiVo* recomenda shows de TV e filmes.

De acordo com [6], os sistemas de recomendações são classificados em três tipos, conforme a maneira como as recomendações são feitas: (i) *recomendações baseadas em conteúdo*: os usuários receberão recomendações de itens similares aos que eles preferiram no passado; (ii) *recomendações colaborativas*: os usuários receberão recomendações de itens que pessoas com gostos e preferências semelhantes têm gostado e (iii) *abordagens híbridas*: combinam as recomendações baseadas em conteúdo e as colaborativas. Zaier et al [12] afirmam que os sistemas de recomendação de filtragem colaborativa (do tipo ii), em particular, têm sido empregado com bastante sucesso.

No contexto deste trabalho, as recomendações são extremamente diretas e feitas com base nas seleções de categoria por parte do usuário. O processo de recomendação consiste basicamente em verificar se o gênero e subgênero dos programas (itens, nos termos de sistemas de recomendação) pertencem ao conjunto de gêneros e subgêneros selecionados pelos usuários. Em caso afirmativo, o programa será recomendado, em caso negativo, o programa não será recomendado.

2.2 Arquitetura da EGA

A arquitetura proposta por [10] é dividida em oito componentes, conforme ilustra a Figura 1. Com exceção do componente *Update Manager*, responsável por realizar as atualizações dos demais componentes, e do componente *EGD Player*, responsável por executar os documentos que compõem a EGA (no nosso contexto, os documentos que compõem a EGA são documentos NCL e scripts em Lua), todos os seis outros componentes podem ser atualizados em tempo real. Essa característica é viabilizada pelo uso de comandos de edição *on-the-fly* de elementos da linguagem NCL em componentes existentes. Os asteriscos (*) das Figuras 1 e 5 indicam os componentes que podem ser atualizados em tempo real.

O processo de geração do EGI é dividido em dois estágios: *feeding* e *production*. No estágio *feeding*, os metadados que serão processados e apresentados pelo EGI são obtidos e transformados em estruturas de dados internas chamadas *Service Data*, as quais são posteriormente passadas para o estágio *production*. No estágio *production*, são gerados os Documentos de Guia Eletrônico (EGDs) que são executados pelo *EGD Player*, normalmente providos pelo middleware, que no *MyPersonal-EPG* é o *Ginga*.

O *feeding* transforma os metadados obtidos por difusão, a partir das emissoras, ou sob demanda, a partir de provedores de conteúdo, em informações relevantes (como, por exemplo, os descritores de gênero e subgênero de um programa) para a EGA que serão armazenadas na estrutura *Service Data*. O componente *Search Engine*, que é controlado pelo componente *Data Manager*, é responsável por coletar essas informações passo a passo (orientado a eventos). Em cada passo, uma vez que a informação é coletada, ela é traduzida e adicionada à

estrutura *Service Data*. Uma característica interessante é que as informações coletadas podem ser descartadas se, por ventura, não sejam consideradas relevantes para exibição.

A estrutura *Service Data* é, basicamente, uma lista de informações acerca dos programas ou serviços em um formato interno que é relevante para a EGA. Ela representa informações referentes aos programas, que podem ainda ser enriquecidas com conteúdo adicional obtido sob demanda pelo componente *Search Engine*. No *MyPersonal-EPG*, a estrutura *Service Data* é basicamente a lista de programas obtida junto às emissoras por difusão ou sob demanda. Cada entrada da lista contém informações sobre um determinado programa como, por exemplo: o nome, o canal e o horário em que será exibido, a duração e o gênero e subgênero. Tais informações são disponibilizadas para o estágio *production* bem como para o componente *History Manager*. A inserção da estrutura *Service Data* na arquitetura provê modularização, já que desacopla os estágios *feeding* e *production*, permitindo a substituição de qualquer componente do estágio *feeding* em tempo de execução

O componente *History Manager* pode fazer uso do componente *Component Persistency* para persistir informações, que serão usadas futuramente por outras aplicações, existentes na estrutura *Service Data*. Aplicações de auditoria e de recomendações de programas podem ser exemplos de aplicações que poderiam fazer uso desse tipo de informação.

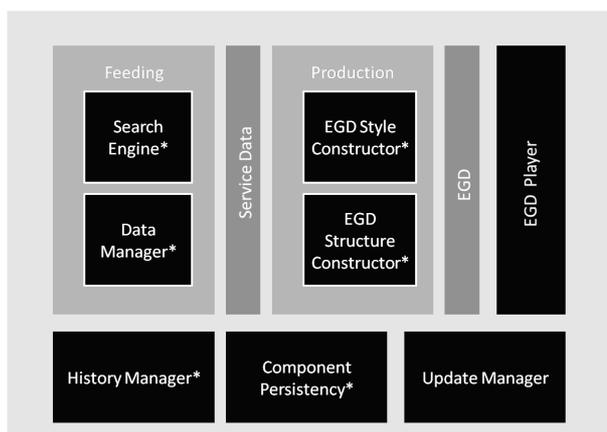


Figura 1. Arquitetura de EGA proposta por [10]

O resultado do estágio *production* é um conjunto de documentos (arquivos) escritos em uma linguagem suportada pelo middleware. No *MyPersonal-EPG*, conforme já mencionamos, tais documentos são escritos em NCL ou em Lua. Os autores preocuparam-se em separar os aspectos de *layout* e de estrutura de apresentação de dados do EGD. O componente *EGD Style Constructor* é responsável por criar os estilos de apresentação que serão usados no momento em que as informações existentes no *Service Data* forem exibidas. O componente *EGD Structure Constructor* é responsável por organizar, estruturar e relacionar os conteúdos a serem apresentados. Na linguagem NCL, o componente *EGD Style Constructor* está relacionado a construtores como: `<region>` e `<descriptor>`, enquanto o componente *EGD Structure Constructor* está relacionado a construtores como: `<media>` e `<link>`.

O EGD é processado pelo *EGD Player* que então exibe o EGI. É importante observar que o próprio *EGD Player* pode realizar adaptações no EGI durante o processamento do EGD.

O componente *Component Persistency* é responsável, como podemos inferir pelo seu próprio nome, pela persistência de informações úteis para a EGA. Tais informações podem ser informações existentes na estrutura *Service Data* ou ainda versões dos componentes da própria EGA.

O componente *Update Manager* é um metasserviço responsável por manter versões *default* de todos os outros componentes (com exceção do *EGD Player*, que é de responsabilidade do middleware) e por receber requisições para atualizá-los em tempo real. O processo de atualização de componente envolve a atividade de validação, a fim de garantir a consistência do sistema depois da atualização. As atualizações também podem ser feitas usando componentes persistentes que já estão presentes no *Update Manager*, ou ainda usando componentes baixados através de alguma rede. Questões de segurança e de direito de acesso também devem ser levadas em consideração pelo *Update Manager*, durante o processo de atualização dos componentes.

Essa arquitetura não contempla mecanismos prover as funcionalidades de criação de grades de programação personalizadas e recomendações. Dessa forma, propomos uma extensão que será descrita em detalhes na seção 3.2.

3. MYPERSONAL-EPG

O EPG proposto nesse trabalho visa atender aos seguintes requisitos: (i) permitir aos usuários a montagem do seu próprio quadro de programação; (ii) oferecer um mecanismo para sintonizar os devidos canais no momento em que os programas selecionados pelo usuário estiverem prestes a começar; (iii) permitir que os usuários selecionem categorias de programas desejadas; (iv) oferecer recomendações, síncronas e assíncronas, de programas com base nas categorias selecionadas previamente pelos usuários; (v) permitir que os usuários modifiquem as opções de configuração; (vi) possibilitar a criação de diversas contas de usuários, a fim de permitir que cada usuário possa armazenar todas as suas informações de interesse.

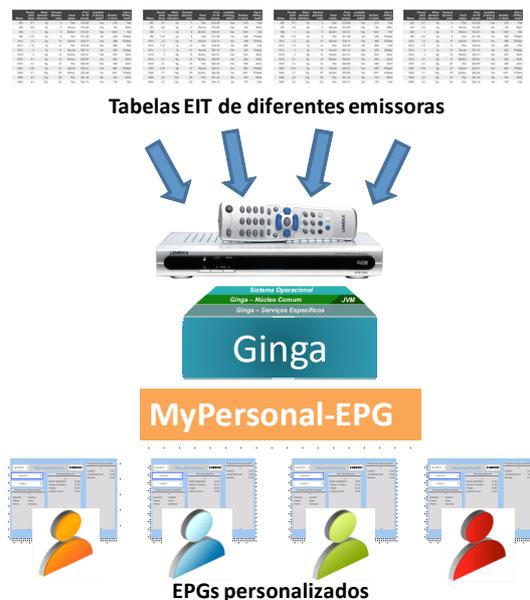


Figura 2. Visão Geral

A Figura 2 ilustra uma visão geral do contexto onde o *MyPersonal-EPG* enquadra-se. Vários EPGs de diferentes emissoras são recebidos por receptores de TV Digital que contêm o middleware Ginga e o *MyPersonal-EPG*. Diversos usuários utilizam o *MyPersonal-EPG* para configurar sua própria grade de programação e registrar seus interesses em termos de programas e recomendações. Dessa forma, os diferentes usuários registrados no *MyPersonal-EPG* podem utilizar a sua EPG personalizada.

3.1 Interface com o Usuário

De forma a permitir a utilização do *MyPersonal-EPG* foi desenvolvida uma ferramenta cuja interface de usuário é descrita nessa seção. Esta interface de usuário foi implementada utilizando as tecnologias previstas no SBTVD e seguindo as diretrizes propostas em [4] e [5]. O SBTVD prevê o uso da linguagem declarativa NCL, da linguagem de script Lua e do *middleware* Ginga no desenvolvimento de aplicações.

A Figura 3 mostra a tela inicial da interface do *MyPersonal-EPG*. A barra de informações exibe ao usuário dados importantes como: data, título e canal. Logo abaixo, há uma tabela que mostra os programas que serão exibidos juntamente com várias informações sobre eles. À direita, há um quadro que mostra quais programas foram selecionados pelos usuários até o momento. Na parte inferior, há a barra de interação que será usado pelos usuários para explorar as funcionalidades da aplicação.



Figura 3. MyPersonal-EPG - Programas disponíveis (esquerda) e Programas Selecionados.

Um dos princípios de design que seguimos durante o desenvolvimento da IU foi o de mostrar ao usuário quais são os mecanismos de interação existentes com cada um dos elementos da tela. A barra de interação, por exemplo, possui 4 retângulos coloridos com texto indicando a ação correspondente sobre eles (um vermelho, um verde, um amarelo e um azul). As cores estão tentando mostrar que para realizar as devidas ações o usuário deve usar os respectivos botões coloridos do controle remoto. As setas mostradas na interface têm a intenção de indicar ao usuário que ele pode usar as setas do controle remoto para interagir com os elementos da interface. No caso da data mostrada na barra de informações, por exemplo, ao utilizar as setas para cima e para baixo do controle remoto, o usuário perceberá que a data será modificada e os programas exibidos na tabela abaixo serão também modificados já que os

programas exibidos estão associados à data que está sendo exibida no momento.

A inserção e remoção de programas da grade de programas selecionados são feitas de maneira bastante simples, a partir do uso das teclas coloridas do controle remoto. O usuário pode navegar entre os programas que estão em exibição usando as setas para cima e para baixo do controle remoto e, ao encontrar o programa desejado, simplesmente apertar a tecla vermelha para inseri-lo em sua grade de programação. Se, em um momento futuro, o usuário desistir de assistir a um programa que pertence à sua grade, basta selecionar o programa na grade e apertar o botão verde para removê-lo.

A borda de seleção branca (na tela inicial o primeiro programa - “Carros” - está selecionado) dá a noção de navegabilidade, que pode ocorrer quando o usuário aperta as setas para cima e para baixo do controle remoto. A única interação que fica implícita é o uso das setas para direita e para esquerda do controle remoto para alternar entre os menus. Existem, conceitualmente, três menus na tela inicial: o primeiro é representado pelos 5 botões laranjas que aparecem à esquerda; o segundo é representado pelo quadro contendo a lista de programas selecionados; e o terceiro é representado pelo quadro contendo a data. A navegação entre os menus é feita de maneira circular. Para ir do *menu* com os botões laranjas para o menu com o quadro de programas selecionados, por exemplo, basta aperta a seta para direita. Apertando o mesmo botão novamente, o *menu* do quadro da data será selecionado. Apertando a seta para direita mais uma vez, o *menu* com os botões laranjas será novamente selecionado. Idéia análoga é empregada com a interação pela seta para esquerda. Em termos de mecanismos de interação, as demais telas (a tela 3 é ilustrada pela Figura 4) são semelhantes à tela 1 (Figura 3). Dessa forma, descrevemos apenas brevemente a tela 3.



Figura 4. MyPersonal-EPG Categorias, Recomendações e Programas Selecionados.

Na tela mostrada na Figura 4, há dois *combo box* brancos responsáveis por permitir aos usuários que eles selecionem as categorias desejadas. O texto escrito em cada *combo box* é modificado pelo uso das setas para cima e para baixo, conforme indicam as setas ao lado. Logo abaixo, há um quadro que exibe quais categorias estão cadastradas no presente momento. No centro da tela, há um quadro de programas recomendados pelo *MyPersonal-EPG* que foram disponibilizados pelos canais após o usuário ter cadastrado as categorias de interesse (modo assíncrono de recomendação). Estes programas podem ser movidos para o quadro de programas selecionados (mostrado à direita) de maneira bastante simples, basta apenas selecionar o

programa na lista de programas recomendados e apertar a tecla vermelha do controle remoto. Uma vez movidos, os programas passam a fazer parte da lista de programas selecionados e a aplicação sintonizará os devidos canais no momento adequado. Todos os quadros presentes nesta tela (quadro de categorias selecionadas, programas recomendados e programas selecionados) permitem, quando estão devidamente selecionados, que o usuário possa navegar pela lista de valores usando as setas para cima e para baixo. O valor atualmente selecionado terá o texto em cor branca para que o usuário possa perceber e podem ser removidos diretamente através da tecla colorida verde.

Do ponto de vista de funcionalidades, a tela da Figura 3 está relacionada aos itens (i) e (ii) apresentados no início desta seção. A tela da Figura 4, por sua vez, aos itens (iii) e (iv). Não esta mostrada aqui a tela de configurações que refere-se ao item (v).

3.2 Arquitetura

A arquitetura do MyPersonal-EPG está ilustrada na Figura 5 e estende a arquitetura de [10] (Figura 1) incluindo dois novos componentes – *Scheduler* e *Recommender* - para atender os requisitos descritos no início da seção 3.

O componente *Scheduler* é responsável por avaliar periodicamente (p.ex., a cada 60 segundos) a lista de programas selecionados pelo usuário e tomar a ação correspondente. Para desempenhar sua função, esse componente precisa ter acesso à estrutura *Service Data*, à data e hora atuais e às informações de configuração do usuário. A cada vez que procede a avaliação, o *Scheduler* verifica se algum dos programas selecionados está prestes a começar, com base na data e hora atuais. Em caso afirmativo, ele deve checar as informações de configuração para o usuário que está usando a aplicação no momento, especificamente a informação que indica se os programas selecionados devem ser automaticamente sintonizados ou não.

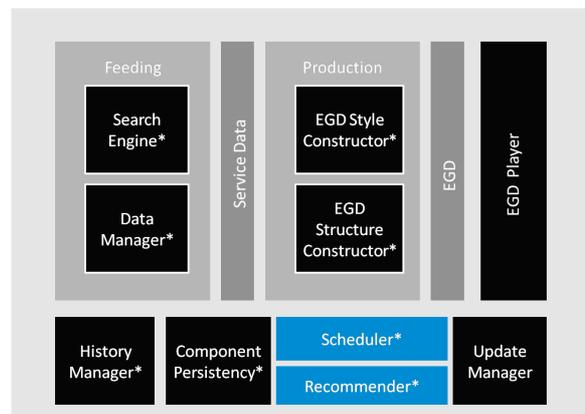


Figura 5 – Arquitetura do MyPersonal-EPG

Se o usuário tiver marcado a opção como “Sim”, o *Scheduler* dispara um evento responsável por sintonizar o canal correspondente. Caso contrário, ele exibe uma mensagem de aviso na tela da televisão, indicando ao usuário que o programa selecionado em questão já está sendo exibido. Se o usuário decidir, em um momento posterior à exibição da mensagem, que deseja assistir o programa, basta apenas apertar a tecla *enter* do controle remoto e o *Scheduler* sintonizará o canal adequado. Se o usuário decidir que não deseja ir para o programa

selecionado e que não deseja mais ver a mensagem de aviso, basta apenas apertar a tecla azul do controle remoto.

Do ponto de vista arquitetural, o componente *Scheduler* usa as informações armazenadas na estrutura de dados *Service Data*, havendo, portanto, dependência indireta em relação aos componentes do estágio *feeding*.

O componente *Recommender* é responsável por fazer recomendações de programas com base nas categorias selecionados pelos usuários. Para tanto, ele precisa ter acesso às informações de categoria do usuário (em termos de gênero e subgênero) e aos dados contidos na estrutura *Service Data*. O *Recommender* analisa os dados da *Service Data* e verifica se os programas devem ser adicionados ao quadro de recomendações. Se o gênero e subgênero do programa for compatível com um dos gêneros e subgêneros selecionados pelo usuário, o programa será adicionado ao quadro de recomendações. Uma vez que são mostrados no quadro de recomendações, os programas podem ser transferidos para o quadro de programas selecionados pela interação do usuário com o controle remoto. O mecanismo de análise que acabamos de descrever é síncrono, já que é disparado sempre que uma nova categoria é inserida ou removida pelo usuário.

Existe, no entanto, a necessidade de garantir que as recomendações sejam atualizadas constantemente, sem que seja preciso para isso o usuário reinserir uma categoria. Dessa forma, desenvolvemos um mecanismo assíncrono para atualizar as recomendações. Este mecanismo é baseado na comunicação entre o componente *Search Engine* e componente *Recommender*. Sempre que recebe novas informações acerca de programas, o *Search Engine* transmite tais informações ao *Recommender* diretamente. De posse dessas informações e das informações de categoria do usuário, o *Recommender* é capaz de atualizar automaticamente o quadro de recomendações. Portanto, o usuário tem apenas de inserir as categorias desejadas uma única vez e as recomendações permanecem sendo atualizadas continuamente.

4. IMPLEMENTAÇÃO DO MYPERSONAL-EPG

A implementação do *MyPersonal-EPG* foi realizada em NCL/Lua. O diagrama de componentes, exibido na Figura 6, apresenta os principais componentes da aplicação, em termos de artefatos de código e arquivos de dados.

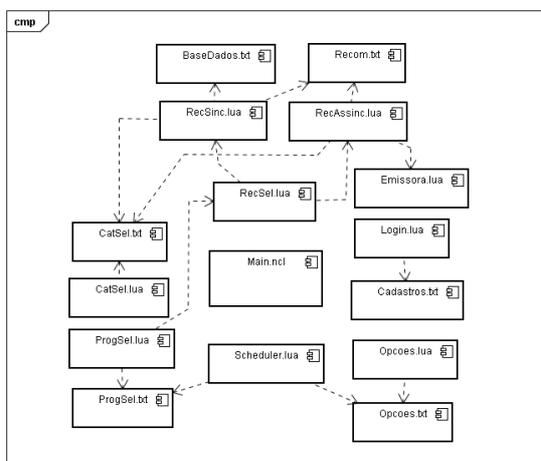


Figura 6. Diagrama de componentes da aplicação

Um dos principais componentes da aplicação é o *Main.ncl*, responsável por: permitir a comunicação entre os demais componentes; estabelecer o posicionamento dos elementos de interface na tela; e definir elos que possibilitam a comunicação da aplicação com o *middleware*. Para não tornar a imagem na Figura 6 demasiadamente carregada de informações, não inserimos as dependências entre os componentes *.lua* e o *Main.ncl*. Podemos considerar que cada componente *.lua* possui dependência em relação ao *Main.ncl*. Alguns outros componentes merecem destaque nessa breve descrição. O componente *Login.lua* tem a função de prover acesso às funcionalidades da aplicação aos usuários. Ao realizar o login, cada usuário tem acesso às suas informações de perfil. O componente *Login.lua* acessa o arquivo *Cadastros.txt* a fim de inserir um novo usuário ou para verificar as informações submetidas em uma tentativa de *login*.

Os componentes *CatSel.lua* e *ProgSel.lua* são responsáveis por gerenciar, respectivamente, as categorias e os programas selecionados pelos usuários. O componente *Opcoes.lua*, por sua vez, tem o papel de gerenciar as configurações dos usuários. Os componentes *RecSel.lua*, *RecSinc.lua* e *RecAssinc.lua* estão relacionados à característica de recomendações provida pela aplicação. O primeiro gerencia o quadro de recomendações que pode ser usado pelos usuários para inserir a recomendação na lista de programas selecionados. Os dois últimos referem-se à obtenção de recomendações de maneira síncrona e assíncrona, respectivamente. O componente *Emissora.lua* foi desenvolvido no intuito de simular o papel das emissoras de conteúdo, responsáveis por enviar eventos indicando a disponibilidade de novos programas. Naturalmente, o componente *RecAssinc.lua* depende de *Emissora.lua*. O componente *Scheduler* é responsável por sintonizar os canais dos programas selecionados pelos usuários no momento adequado e de acordo com as configurações estabelecidas. Mais detalhes em relação ao funcionamento dos componentes de recomendação e do componente *Scheduler.lua* serão vistos a seguir.

O trecho de código apresentado na Figura 7 ilustra o funcionamento do componente *Scheduler*. Alguns detalhes foram omitidos por questões de espaço. Na linha 1, o arquivo *ProgSel.txt*, que contém os programas selecionados pelo usuário, é aberto. O arquivo contém uma linha de texto para cada programa selecionado. Nesta linha estão escritos o nome do programa, o canal em que ele será exibido e o horário em que ele vai começar, em termos de horas e minutos (e eventualmente data). No laço que se inicia na linha 3 e termina na linha 16, a aplicação faz a leitura do arquivo, linha a linha. Na linha 4 é feito o *parsing* de cada linha do arquivo. O *parsing* é responsável por carregar a estrutura *pro* com os dados do programa selecionado. O atributo *pro.name* representa, portanto, o nome do programa tratado no momento. Na linha 8, a aplicação checa se a hora e o minuto do programa (*pro.hour* e *pro.min*, respectivamente) correspondem a hora e o minuto atual (*hour* e *min*, respectivamente). Se isto for verdade, a aplicação continua sua execução a partir da linha 9. Caso contrário, a aplicação itera o laço da linha 3 e passar a analisar a próxima linha do arquivo (se houver). Na linha 10, o arquivo de configuração *Opcoes.txt* é aberto. Esse arquivo contém informações sobre as configurações da aplicação escolhidas pelos usuários. Neste trecho de código, somente a opção de configuração “Ir automaticamente para os programas selecionados” é importante. A aplicação checa se essa opção

está com o valor “Sim” ou “Não”. A aplicação usará tal informação para tomar a decisão correspondente. A partir da linha 14 (código omitido por razões de espaço) a aplicação sintoniza o canal correspondente imediatamente, caso a opção de configuração “Ir automaticamente para os programas selecionados” tenha o valor “Sim”. Caso contrário, a aplicação exibirá uma mensagem na tela, indicando ao usuário que o programa em questão já começou e que ele pode ir para o canal que está exibindo o referido programa apertando a tecla ENTER do controle remoto.

```

1: local file = io.open("ProgSel.txt",'r')
2: if file ~= nil then
3:   for line in file:lines() do
4:     parseProg(line)
5:     local j = string.find(pro.starttime,":")
6:     local proghour = tonumber(string.sub(pro.starttime,1,j-1))
7:     local progmin = tonumber(string.sub(pro.starttime,j+1))
8:     if proghour == hour and progmin == min then
9:       local type = ''
10:      local file3 = io.open("Opcoes.txt",'r')
11:      if file3 ~= nil then
12:        -- testa se mudança de canal é automática
13:      end
14:      -- código para ir para o canal correspondente
15:    end
16:  end
17: end

```

Figura 7. Trecho de código do componente Scheduler

O trecho de código apresentado na Figura 8 ilustra a implementação do componente *Recommender*. Nas linhas 1 a 31, está definida a função responsável por receber eventos por parte do *middleware*, mais especificamente eventos do tipo SI (*Service Information*). Na linha 32, a função é registrada usando o módulo *event*, que faz parte do Ginga. Na linha 2, a aplicação verifica se o evento recebido é efetivamente um SI. Em caso positivo, o processamento continua na linha 3. Na linha 3, o evento recebido é transformado em uma estrutura chamada *nv*, que será usada posteriormente. Na linha 4, a aplicação abre o arquivo *CatSel.txt* que contém a lista de categorias selecionadas pelo usuário, que será usada posteriormente para verificar se o evento recebido deve ou não ser inserido na lista de recomendações do usuário. No laço que se inicia na linha 6, a aplicação insere o evento recebido na lista de recomendações do usuário caso se enquadre nas categorias escolhidas (encontradas no arquivo *CatSel.txt*). O referido laço itera sobre o conjunto de categorias selecionadas, verificando se o evento recebido corresponde à categoria selecionada em análise.

Nas linhas 7 a 9, a aplicação faz o *parsing* das categorias, obtendo o gênero e o subgênero. A condição da linha 10 verifica se o evento recebido enquadra-se na categoria atualmente em análise. Em caso positivo, a execução da aplicação continua na linha 11. Nas linhas 12 a 14, a aplicação verifica se o evento recebido já faz parte da lista de recomendações do usuário. Se isso for verdade, não será necessário inseri-lo novamente. Caso contrário, a aplicação abre novamente o arquivo *Recom.txt* (desta vez para escrita) e insere o programa na lista de recomendações do usuário (linhas 15 a 37). Nas linhas 18 a 20, a aplicação envia um evento (de nome *newrec*) que indica à interface para se atualizar, já que uma nova recomendação foi inserida. Finalmente, na linha 28, a aplicação indica ao *middleware* que terminou de tratar o evento e está pronta para receber novos eventos. Vale lembrar que o trecho de código refere-se ao mecanismo assíncrono do componente *Recommender*. Esse mecanismo garante que as

recomendações sejam exibidas ao usuário à medida que são enviadas pela emissora. O mecanismo síncrono, por sua vez, faz com que, sempre que o usuário insira uma nova categoria, a aplicação busque por programas previamente armazenados em uma base de dados e os exiba ao usuário.

5. TRABALHOS RELACIONADOS

Em [13], os autores propõem a criação de um Guia Eletrônico de Programação Pessoal com base na definição de perfis de usuários e na descrição dos programas. Os autores alegam que a partir de tais informações é possível produzir um EPG contendo apenas programas relevantes para os usuários, o que evita que usuários precisem navegar entre os canais ou ler os guias de programação. Os autores mencionam que os perfis dos usuários podem ser criados e atualizados de maneira automática através de algoritmos de *clustering*. Essencialmente, eles aplicam algoritmos de casamento (*matching*) entre os perfis dos usuários e as descrições dos programas para produzir um guia de programação com as preferências dos usuários. Comparando com o *MyPersonal-EPG*, o trabalho de [13] tem um escopo centrado em desenvolver algoritmos de *matching* e de *clustering*, que não fez parte dos objetivos do nosso trabalho. Por outro lado, o *MyPersonal-EPG* tem um escopo mais amplo, visando atender vários requisitos básicos dos usuários de um EPG, além das recomendações.

```

1: function handler (evt)
2:   if evt.class=='ncl' and evt.type=='attribution' and evt.name=='SI_Event'
3:   then
4:     nv = stringtoTable(evt.value)
5:     file = io.open("CatSel.txt", 'r')
6:     if file ~= nil then
7:       for line in file:lines() do
8:         j = string.find(line,":")
9:         genero = string.sub(line,1,j-1)
10:        subgenero = string.sub(line,j+1,line:len())
11:        if nv.genero == genero and nv.subgenero == subgenero then
12:          local ins = true
13:          file2 = io.open("Recom.txt",'r')
14:          if file2 ~= nil then
15:            -- verifica se o evento faz parte das recomendações
16:            if ins == true then
17:              file2 = io.open("Recom.txt",'a')
18:              if file2 ~= nil then
19:                event.post {class='ncl',type='attribution',name='newrec',
20:                           action='start',value=evt.value,}
21:              end
22:            end
23:          end
24:        end
25:      end
26:      file:close()
27:    end
28:    event.post {class='ncl',type='attribution',name='SI_Event',
29:               action='stop',value=evt.value,}
30:  end
31: end
32: event.register(handler)

```

Figura 8. Trecho de código do componente Recommender

Em [10], os autores definem uma arquitetura de referência para aplicações de EPG sensíveis ao contexto da apresentação. O trabalho apresenta os componentes da arquitetura, suas respectivas funções e o relacionamento entre eles. Os autores apresentam também trechos de código dos componentes referentes à implementação de referência em NCL-Lua. No

nosso trabalho, estendemos a arquitetura proposta por [10], acrescentando os componentes *Scheduler* e *Recommender*, descritos nas seções anteriores. Desenvolvemos também a implementação de um EPG com funcionalidades adicionais e muito úteis como a seleção e recomendação de programas, além da funcionalidade essencial de um EPG: a exibição da lista de programas em exibição ou que ainda serão exibidos.

O trabalho [14] propõe um sistema de recomendações de programas de TV baseado em uma abordagem de mineração de dados. A implementação foi desenvolvida como uma extensão do Ginga. Diferente do *MyPersonal-EPG*, a abordagem proposta não faz uso de informações diretamente providas pelos usuários, somente da análise histórica dos programas vistos por eles. Também não faz sintonização automática de programas. Um dos problemas citados pelos autores é o tempo de processamento necessário ser elevado. Para atenuar o problema, eles sugerem que o processamento seja feito em períodos em que o STB (*set top box*) esteja ocioso. Outro problema é a necessidade que a aplicação tem de “aprender” de que tipo de programa o usuário gosta. Alguns usuários podem não estar dispostos a usar a aplicação até que ela finalmente passe a fazer boas recomendações.

6. CONCLUSÕES

Neste artigo apresentamos a arquitetura, implementação e interface do usuário do *MyPersonal-EPG*, que provê, além das funcionalidades básicas de um EPG comum (navegação em grades de programas e seleção de canais), as funcionalidades de montagem de quadro de programação personalizado e de provisão de recomendações com base em categorias escolhidas pelos usuários. Além disso, o *MyPersonal-EPG* pode sintonizar os programas selecionados automaticamente no momento adequado, sem que o usuário tenha que se preocupar em efetuar a mudança de canal. A aplicação também permite a criação de perfis de usuários, de maneira que cada usuário que tenha acesso ao STB possa gerenciar suas próprias informações (programas e categorias selecionadas, recomendações e configurações da aplicação).

O *MyPersonalEPG* preenche uma lacuna importante no contexto atual da TV digital, onde há uma diversidade de conteúdos digitais disponíveis e há necessidade de ferramentas para: (i) auxiliar o usuário a organizar sua própria programação, (ii) informar ao usuário a disponibilização de programas do seu interesse e (iii) avisar ao usuário que um programa selecionado está começando ou sintonizá-lo automaticamente, de forma a evitar que o usuário perca a exibição de um programa.

A aplicação foi desenvolvida e testada em um ambiente simulado (a máquina virtual GINGA-NCL VIRTUAL STB, disponibilizada em [16]). Os programas de TV foram simulados por vídeos armazenados no próprio sistema de arquivos e a aplicação não recebia nenhuma informação das emissoras (o que motivou a criação do componente *Emissora.lua* para testar o comportamento do componente de recomendação assíncrona). A troca de canais era simulada pela troca de vídeos (armazenados no sistema de arquivos) em exibição.

Em termos de trabalhos futuros, novas funcionalidades são necessárias, especialmente no mecanismo de interação.

Mecanismos de busca mais sofisticados podem auxiliar o usuário, p.ex., busca por atributos específicos de acordo com a classe do programa, tais como por atores e diretores (no caso de filmes), por time (no caso de futebol). Além disso, a aplicação deve permitir que o usuário indique que deseja sempre assistir a um programa que é exibido periodicamente (uma série de TV, por exemplo).

7. REFERÊNCIAS

- [1] ABNT NBR 15603-1 (2007) – Digital Terrestrial Television – Multiplexing and Service Information (SI) Part 1: SI for Digital Broadcasting Systems, ABNT Standard, November, ISBN 978-85-07-00610-7, ABNT NBR 15603-1:2007.
- [2] ABNT NBR 15603-2 (2007) – Associação Brasileira de Normas Técnicas, “Televisão digital terrestre – Multiplexação e serviços de informação (SI) Parte 2: Estrutura de dados e definições da informação básica de SI”, SBTVD Terrestre, NBR 15603-2.
- [3] ABNT NBR 15606-2 (2007) – Associação Brasileira de Normas Técnicas, “Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações”, Sistema Brasileiro de TV Digital Terrestre, NBR 15606-2.
- [4] Barbosa, S. D. J. e Soares, L.F.G. (2008). TV digital interativa no Brasil se faz com Ginga: Fundamentos, Padrões, Autoria Declarativa e Usabilidade. Rio de Janeiro, RJ.: <http://www.ncl.org.br/documentos/JAI2008.pdf>.
- [5] British BroadCasting Corporation (2002) – “Interactive Television Guide”.
- [6] Adamovicus, G. e Tuzhilin, A. (2005) “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions”, Transactions on Knowledge and Data Engineering, vol. 17, pp. 734-749
- [7] Hjelm, J. (2008) Why IPTV: Interactivity, Technologies, and Services, John Wiley and Sons Ltda.
- [8] Ierusalimsky, R. (2008) Programming in Lua, Lua.org.
- [9] ISO/IEC 13818-1 (2000) International Organization for Standardization / Internacional Eletrotecnical Committee, “Information Technology – Generic coding of moving pictures and associated audio information – Part 1: Systems”, ISO/IEC 13818-1.
- [10] Moreno, M. F. et al. (2009) Adaptable software components in an electronic program/service guide application architecture for context aware guide presentation. International Journal of Advanced Media and Communication, v. 3, n. 4, p. 351-364.
- [11] Morris, S. et al. (2005) Interactive TV Standards, Elsevier Inc.
- [12] Zaier, Z.; Godin, R. e Faucher, L. (2008) Evaluating Recommender Systems. In: Automated solutions for Cross Media Content and Multi-channel Distribution, 2008.
- [13] Ehrmantraut, M.; et al. (1996) The Personal Electronic Program Guide – Towards the Pre-selection of Individual TV Programs. In: Proc. 5th Int. Conf. on Information and knowledge management
- [14] Ávila, P. M. e Zorzo, S. D. (2009) A personalized TV Guide System An approach to Interactive Digital Television. In: Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics.
- [15] Tvoon Media Center (2010). Disponível em: <http://www.tvoon.com/index.php>. Acesso em: 8 de maio de 2010.
- [16] Ginga-NCL. Ferramentas Ginga-NCL. Disponível em: <http://www.gingancncl.org.br/ferramentas.html>. Último acesso em: 8 de maio de 2010.