

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE PÓS-GRADUAÇÃO
ESPECIALIZAÇÃO EM DESENVOLVIMENTO WEB

PEDRO HENRIQUE DE SOUSA

**SMART TVS: UMA ANÁLISE SOBRE AS FORMAS DE
DESENVOLVIMENTO DE INTERFACES INTERATIVAS**

MONOGRAFIA DE ESPECIALIZAÇÃO

LONDRINA
2013

PEDRO HENRIQUE DE SOUSA

**SMART TVS: UMA ANÁLISE SOBRE AS FORMAS DE
DESENVOLVIMENTO DE INTERFACES INTERATIVAS**

Monografia apresentada como requisito parcial para obtenção do grau de Especialista em Desenvolvimento Web, do Programa de Pós-Graduação em Desenvolvimento Web, da Universidade Tecnológica Federal do Paraná, campus Londrina. Área de Concentração: Sistemas de Informação.

Orientador: Prof. Me. THIAGO P. DE CAMPOS.

LONDRINA
2012

TERMO DE APROVAÇÃO



Título da Monografia

**SMART TVS: UMA ANÁLISE SOBRE AS FORMAS DE
DESENVOLVIMENTO DE INTERFACES INTERATIVAS**

por

PEDRO HENRIQUE DE SOUSA

Esta monografia foi apresentada às 16h00 do dia **23** de **fevereiro** de 2013 como requisito parcial para a obtenção do título de ESPECIALISTA EM DESENVOLVIMENTO WEB. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Thiago Prado de Campos
(UTFPR)

Prof. César Cusin
(FAP-CE)

Prof. Reinaldo Ferraz
(W3C)

Visto da coordenação:

Prof. Thiago Prado de Campos
Coordenador da esp. em Desenvolvimento Web

Prof. Walmir Eno Pottker
Coordenador de Pós-Graduação Lato Sensu

À minha família !

AGRADECIMENTO

Agradeço a Me. Thiago P. de Campos pela orientação desta pesquisa e pelos momentos de aprendizado. Agradeço, também, aos colegas de classe, os quais sempre estiveram dispostos a compartilhar conhecimentos e experiências vividas, contribuindo para o melhor desenvolvimento deste trabalho.

E não poderia deixar de agradecer à minha família, pois acredito que sem o apoio deles seria muito difícil vencer esse desafio. E por último, e nem por isso menos importante, agradeço a minha namorada pelo carinho, amor e compreensão.

A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original. (*Albert Einstein*).

RESUMO

SOUSA, Pedro H. **SMART TVS**: Uma análise sobre as formas de desenvolvimento de interfaces interativas. 59f. 2012. Monografia (Especialização em Desenvolvimento Web) – Universidade Tecnológica Federal do Paraná. Londrina, Paraná, 2013.

Com a grande evolução tecnológica que estamos vivenciando, e uma necessidade cada vez maior de estar conectado a internet, seja navegando por sites de notícias, entretenimento, redes sociais ou mesmo estudando, não demorou muito para que todas estas vantagens saíssem da tela do computador e fossem para também em nosso celulares, dando maior mobilidades para os usuários. E seguindo este mesmo caminho ao iniciar o processo de modernização de uma das maiores invenções do homem, as televisões se modernizaram e também passara a oferecer o recurso de conectividade com a internet. Este trabalho busca realizar uma pesquisa para levantar dados sobre os novo modelo de equipamento que esta se destacando no mercado chamados de *Smart TV*, para compreender melhor o seu funcionamento e realizar um levantamento dos principais recursos oferecidos, referentes a forma de transmissão e interação com o Usuário, propondo assim formas de se construir aplicativos para estes, demonstrando as possibilidades que estes aparelhos podem oferecer propondo possíveis formas de interação que poderiam ser aplicadas na construção de interfaces direcionadas a um âmbito esportivo, demonstrando assim algumas das possibilidades que estes equipamentos podem oferecer tanto para usuários quanto para emissoras de televisão. Ao final será apresentado as telas de um protótipo o qual foi elaborado para aplicar os conceitos de interação propostos no trabalho, demonstrando como seria possível a criação destas interfaces.

Palavras-Chave: Smart TV. Interfaces interativas. Desenvolvimento de aplicativos. Transmissão esportiva.

RESUMO

SOUSA, Pedro H. **SMART TVS**: An analysis on ways of developing interactive interfaces. 59f. 2012. Monografia (Especialização em Desenvolvimento Web) – Universidade Tecnológica Federal do Paraná. Londrina, Paraná, 2013.

With the great technological developments we are experiencing, and a growing need to be connected to the Internet, either by browsing news sites, entertainment, social networking or even studying, not long before all these advantages come out of the computer screen and also to be on our phones, giving greater mobility to users. And following this same path to begin the process of modernization of one of the greatest inventions of man, the televisions were modernized and also come to offer the facility of internet connectivity. This work aims to conduct a search for collecting data on the new model of equipment that stood out on the market called Smart TV, to better understand its operation and conduct a survey of the main features offered, regarding the mode of transmission and interaction with the User, thus proposing ways to build these applications, demonstrating the possibilities that these devices can offer proposing possible ways of interaction that could be applied in the construction of interfaces that target a sporting context, thereby demonstrating some of the possibilities that these devices can offer both for individuals and for television stations. At the end of the screens will be presented a prototype which was designed to apply the concepts of interaction in the proposed work, showing how it would be possible to create these interfaces.

Palavras-Chave: Smart TV. Interactive interfaces. Application Development. Sports broadcasting.

LISTA DE TABELAS

Tabela 1 - Especificação Smart TV Samsung	22
Tabela 2 - Tecnologias suportadas pela Smart TVs Samsung.....	23
Tabela 3 - Eventos do objeto BroadCast da Smart TV LG	28
Tabela 4 - Eventos de reconhecimento de voz da Smart TV LG.....	29
Tabela 5 – Requisitos funcionais do sistema	33
Tabela 5 – Requisitos não funcionais do sistema	33

LISTA DE FIGURAS

Figura 1 - Lista de utilização das redes sociais em <i>Smart TVs</i>	10
Figura 2 - Controle da <i>Smart TV Panasonic touchpads</i>	17
Figura 3 - Controle que simula Mouse na TV da LG	18
Figura 4 - Formas de interação Smart TV Samsung	18
Figura 5 - Imagem de transmissão de TV	21
Figura 6 - Transmissão da Smart TV	22
Figura 7 - Interface do SDK LG	24
Figura 9 - Interface de desenvolvimento da Samsung	25
Figura 10 - Emulador 2012 da Samsung.....	26
Figura 11 - Função da API LG para buscar informações de um programa exibido.....	26
Figura 12 - Função da API LG apresentar informações do canal.....	27
Figura 13 - Métodos responsáveis por alterar o canal de transmissão	27
Figura 15 – Acesso ao aplicativo.....	35
Figura 16 - Tela da área de trabalho do aplicativo	36
Figura 17 – Lista de canais	37
Figura 18 - área de vídeos	38
Figura 19 - Comentários em vídeos	40
Figura 20 - Escolha De Câmera	41
Figura 21 - seleção de áudio.....	42
Figura 22 - Estatísticas da partida.....	43
Figura 23 - Tela do replay da partida	45

SUMÁRIO

1 INTRODUÇÃO	9
2 SMART TVS	12
2.1 APLICATIVOS E TECNOLOGIA.....	12
2.2 COMERCIO ELETRÔNICO	13
2.3 FABRICANTES E SUPORTE	14
2.4 ALTERNATIVAS AS SMART TVS	15
2.5 SMART TVS E SUAS FORMAS DE INTERAÇÃO	17
2.6 TECNOLOGIA DLNA.....	19
3 ARQUITETURA DAS SMART TVS	21
3.1 ARQUITETURA.....	21
3.2 FERRAMENTAS E APIS	24
4 ESCOPO DO PROJETO	31
4.1 REQUISITOS DO SISTEMA.....	31
4.1.1 Requisitos Funcionais	32
4.1.2 Requisitos não funcionais	33
4.2 COLETA DE DADOS	34
4.3 CENÁRIOS.....	34
4.3.1 Cenário 1 – acessando o aplicativo	35
4.3.2 Cenário 2 – área de trabalho do aplicativo	36
4.3.3 Cenário 3 – lista de canais	37
4.3.4 Cenário 4 – área de vídeos do aplicativo	38
4.3.5 Cenário 5 – informações sobre campeonatos e clubes	38
4.4 RECURSOS DE VIDEOS E TRANSMISSÕES	39
4.4.1 Comentar vídeo ou transmissão	39
4.4.2 Câmera	40
4.4.3 Narração	41
4.4.4 Informação da partida	42
4.4.5 Enquetes	43
4.4.6 Promoções e vendas	43
4.4.7 Previsão de resultados	44
4.4.8 Replay	44
5 CONCLUSÃO	46
REFERÊNCIAS	48

1 INTRODUÇÃO

Com o grande impulso tecnológico que estamos vivenciando, um dos itens mais comuns na maioria dos lares vem se transformando de uma forma surpreendente. Com telas cada vez mais finas, uma realidade visual incrível, e funcionalidades antes inimagináveis, as televisões voltam a ter destaque principal nos lares de muitas famílias. Isso por que os fabricantes têm produzido equipamentos que vem sendo chamados de *Smart TVs*, ou TVs inteligentes trazendo consigo diversos recursos que permitem ao telespectador, interagir das mais variadas formas, utilizando, desde movimentos corporais a comandos de voz, por exemplo. Assim como os já conhecidos *Smartphones*, as *Smart TVs* oferecem a possibilidade de instalação de aplicativos o que torna possível ter no aparelho uma variada gama de opções de entretenimento, informação e utilitários.

Com a possibilidade de se conectar à internet através da televisão, cria-se uma nova expectativa de mercado onde podemos imaginar telespectadores muito mais participativos nas programações. Juntamente com o aumento na participação destes telespectadores surge uma possibilidade de se estabelecer negócios através destes meios. Durante uma transmissão esportiva, por exemplo, seria muito cômodo se da poltrona de sua sala, o telespectador pudesse visualizar e comprar o novo uniforme de seu clube, enquanto assiste a sua partida, e dentro de algum tempo receberia este em sua casa, tudo isso sem interromper o que estava fazendo. Da mesma forma poderia estar realizando o pedido de uma pizza para acompanhar sua partida de futebol.

Um dos recursos mais utilizados por usuários de internet no Brasil, as redes sociais, também são apontadas com destaque em uso por outros países emergentes através das *Smart TVs*. Segundo pesquisa realizada pelo Gfk¹ os telespectadores de países como Índia e Brasil utilizariam muito mais as redes sociais em suas *Smart TVs* do que telespectadores de outros países como Reino Unido, Estados Unidos e Alemanha. A pesquisa ainda mostra que telespectadores de países Orientais em crescimento, apresentam maior facilidade em utilizar características sociais das *Smart TVs*, enquanto telespectador de nações mais ricas

¹ *GfK*: Grupo alemão de pesquisa de mercado, <http://www.gfk.com>.

ainda existiria uma mentalidade mais analógica, usufruindo pouco dos novos recursos oferecidos pelas *Smart TVs* (DIGITAL_NEWS, 2012).

Uso de TV Conectada	
China	44%
Coréia do Sul	18%
Índia	17%
Brasil	14%
Turquia	13%
Reino Unido	11%
EUA	11%
México	11%
Espanha	8%
Alemanha	8%
Bélgica	6%
Rússia	5%
Holanda	5%

Figura 1 - Lista de utilização das redes sociais em *Smart TVs*.

Questionado sobre o uso de programas que ofereçam alguma forma de interação 42% dos entrevistados brasileiros demonstraram interesse nesta possibilidade, ficando atrás de China com 61% e Índia com 59%. Já dos Americanos apenas 18% demonstraram este interesse, seguido por Reino Unido com 16% e Alemanha com 15% (GFK, 2012).

O pouco tempo de existência destas novas TVs, abre o caminho para que possam ser estudadas as melhores formas de se criar interfaces de aplicações que venham ao encontro de atender as necessidades do usuário no contexto de uso de um televisor. Esta tarefa pode ser encarada como uma forma de construir aplicativos que sejam de fácil aprendizado e possuam uma interface que agrade ao usuário, permitindo que este possa usufruir de seu equipamento sem passar por transtornos: como o de não saber utilizar um aplicativo ou; demorar muito tempo para realizar uma tarefa.

Nosso trabalho objetiva direcionar o entendimento das possibilidades oferecidas por estes novos equipamentos, e testar os recursos oferecidos na criação de aplicativos para *Smart TVs*.

Este mercado ainda não apresenta muitos padrões claramente definidos para métodos de interação oferecidos pelos equipamentos, porém algumas características são adotadas pelos modelos já em uso. Um ponto aparentemente consolidado e que tem envolvido grandes discussões da área trata das tecnologias

adotadas para o desenvolvimento nestes novos equipamentos. O W3C, referência em padronização de desenvolvimento em diversas áreas da *Web*, vem realizando conferências e coordena grupos de discussões para tentar elaborar padrões a serem seguidos pelos fabricantes destes novos equipamentos.

Como forma de direcionar e exemplificar as técnicas aqui utilizadas adotamos como modelo de estudo a construção de interfaces para aplicativos que permitam ao telespectador interagir com transmissões no âmbito esportivo, analisando e tratando características específicas a estes tipos de transmissões.

Desta forma foram pesquisados os principais equipamentos que se enquadram no modelo proposto. Os dados coletados incluem tecnologias utilizadas, e recursos disponíveis para desenvolvimento de aplicações. Em seguida, coletamos e analisamos informações sobre aplicativos já existentes, identificando características que possam ser estabelecidas como padrões deste seguimento, possibilitando assim o desenvolvimento de um modelo conceitual, para a construção de um protótipo para transmissões esportivas.

No capítulo 2 iremos explicar os conceitos relacionados ao ambiente de *smartTVs*. No capítulo 3 mostraremos de forma detalhada a arquitetura de uma *SmartTV*. No capítulo 4 apresentaremos nosso projeto. No capítulo 5 faremos nossas considerações finais.

2 SMART TVS

As Smart TVs também conhecidas como televisores inteligentes trazem como uma de suas principais novidades o fato de poderem se conectar à internet podendo utilizar um cabo de rede padrão como os que utilizamos para conectar um computador, ou através de dispositivos de conexão sem fio (Wi-Fi²). O fato de conectar sua TV a internet por si só já é uma novidade que torna este aparelho mais interessante aos olhos de seus consumidores, porém um fator que as diferencia de suas antecessoras e que com certeza abriu um leque de possibilidade para um novo mercado de aplicações é que estas possuem sistemas operacionais bem semelhantes aos sistemas dos *Smartphones* atuais, assim torna-se possível a instalação de aplicativos que deixam o aparelho mais completo. Serviços oferecidos antes somente para PCs como assistir filmes, series e vídeos por streaming, agora contam com aplicativos para que possam ser acessados de sua *Smart TV*. São eles: *Netflix* (<https://www.netflix.com>), *Plus Hulu* (<http://www.hulu.com>) , *YouTube* (<http://www.youtube.com>), e *Vimeo* (<http://vimeo.com/>).

Além dos aplicativos padrões que as *Smart TVs* trazem consigo, elas oferecem também a possibilidade de baixar outros aplicativos de lojas *on-line*. (Kovach, 2010)

2.1 APLICATIVOS E TECNOLOGIAS

Aplicativos são *softwares* desenvolvidos para rodar em dispositivos específicos como PDAs, Smartphones, telefones celulares, ou TVs. Até então os aplicativos eram geralmente desenvolvidos com foco no dispositivo em que ele seria utilizado, podendo usar linguagem C, C++, .net e J2ME (*Java 2 Micro Edition*). O desenvolvimento para cada tipo de dispositivo pode mudar muito de fabricante para fabricante e de modelo para modelo (Aplicativos..., 2012)

Com a expansão cada vez maior da internet, aplicativos baseados na *Web* começaram a se popularizar. Os aplicativos *Web* são utilizados diretamente

² Wi-Fi: Expressão sinônimo para a tecnologia IEEE 802.11, que permite a conexão entre diversos dispositivos sem fio.

através de um navegador e permitem realizar diversas tarefas, como criar documentos, editar fotos, ouvir músicas, sem a necessidade de instalar um *software* em seu computador. Serviços comumente utilizados como Gmail, Google Maps, são exemplos de aplicativos *Web*. (Sobre..., 2012)

Com o novo mercado de televisores, novas tecnologias também estão sendo adicionadas na construção de aplicativos, como este equipamento tem como principal diferencial a conexão com a internet, nada mais adequado do que adicionar a estes, a possibilidade de desenvolver aplicativos utilizando as mesmas linguagens já utilizadas na internet. Existe no mercado diversos modelos de equipamentos cada um com suas peculiaridades. Alguns equipamentos, já suportam tecnologias HTML5, CSS3, JavaScript e plug-ins Flash. Estas tecnologias já são solidamente utilizadas na construção de páginas Web e agora podem ser utilizadas também na construção de aplicativos para *Smart TVs* (samsungdforum).

Os recursos oferecidos pela linguagem HTML5 torna possível realizar uma gama de tarefas antes encaradas de uma forma penosa, pois dispendiam de grande esforço para sua realização, a construção de formulários, com campos adequados ao preenchimento, como datas e valores numéricos que precisam ser validados, ou mesmo acionando funcionalidades do dispositivo como teclados virtuais ou lista de contatos.

O desenvolvimento de aplicativos para *Smart TVs*, tem em tecnologias como JavaScript, uma forma de disponibilizar APIs, que facilitem a construção de aplicativos pelos desenvolvedores, pois através destas tecnologias os fabricantes podem oferecer funcionalidades que permitem interagir diretamente com os equipamentos de forma mais simples. Através destas APIs é possível acessar informações do aparelho, como canais, volume e utilizando métodos existentes na API, podemos alterá-las.

2.2 COMERCIO ELETRÔNICO

Outro conceito que convém ao se tratar do uso de internet *Smart TVs*, é o comércio eletrônico. Este recurso permite que os canais de vendas levem para

dentro da casa do telespectador toda a loja, permitindo a este realizar uma compra sem precisar ao menos ligar o telefone.

O direcionamento de vendas que pode ser alcançado com aplicativos neste segmento é amplo, podendo ser explorado comercialmente, aproveitando o mercado gerado pelo uso das *Smart TVs*. Direcionar uma propaganda a um público específico que se identifica com uma determinada mercadoria é algo que pode aumentar as possibilidades de conversões em venda. Desta forma as empresas de transmissão poderiam oferecer um novo tipo de serviços de propaganda.

2.3 FABRICANTES E SUPORTE

A empresa Samsung criou uma área de suporte e informações para desenvolvedores que desejam entrar no mercado de desenvolvimento de aplicativos para *Smart TVs*, e oferece em seu fórum, material que auxilia no desenvolvimento de aplicativos, bem como ferramentas para desenvolvimento destes. Realizando o *download* do SDK³ da Samsung, é possível desenvolver e emular⁴ aplicativos para os televisores desta marca.

Tendo isto como uma tendência de mercado não demorou, para que a LG Electronics criasse também seu fórum de discussões para auxiliar os desenvolvedores que desejam criar aplicativos para suas TVs. A LG também fornecesse *download* dos softwares necessários para desenvolvimento e testes de aplicativos.

A empresa Philips juntamente com a LG e a Sharp, oficializaram uma parceria para o desenvolvimento de aplicativos para TVs sobre uma mesma plataforma, o que significa que aplicativos construídos para rodar em TVs da marca LG também funcionarão em equipamentos da Philips. Isto facilita aos desenvolvedores que podem dispensar a necessidade de construir diversas versões do mesmo aplicativo para fabricantes diferentes. Existe possibilidade de outras empresas como a Sony e Panasonic, que também atuam neste mercado, se unirem

³ *Software Development Kit*, ou Kit de Desenvolvimento de Software ou Kit de Desenvolvimento de Aplicativos.

⁴ Reproduzir as funções de um determinado ambiente, a fim de permitir a execução de outros softwares sobre ele.

a esta iniciativa, chamada *Alliance Smart TV*. A expectativa é que mais 25 parceiros façam parte do grupo (Ig, 2012). Uma das principais vantagens para os desenvolvedores e telespectadores é que, com esta aliança, o desenvolvimento de aplicativos será focado em soluções livres, tendo a HTML5 como base, permitindo que empresas interessadas produzam seus próprios aplicativos. Outra característica dos equipamentos fabricados pelas empresas desta aliança é que todos possuirão um navegador embutido que suporta as tecnologia especificadas de modo que o telespectador poderá utilizá-lo para acessar a *Web* sem nenhum problema (Developers, 2012).

A criação de uma aliança nas plataformas de desenvolvimento de aplicativos mostra que os fabricantes de televisores entenderam que a produção de novos aplicativos, gerando uma variedade e estimulando os telespectadores de baixarem e instalarem em seus equipamentos é a única forma de popularizar as *Smart TVs*, da mesma forma como aconteceu com os *smartphones*. A ideia desta aliança é ambiciosa. Se olharmos para o passado e lembrarmos de como começou a evolução dos *smartphones*, com a Apple, depois a Google com o *Android* seguido pela RIM com o *Blackberry* e pela Microsoft. Isto que os fabricantes de televisores estão se propondo, de padronizar o desenvolvimento de aplicativos utilizando tecnologias abertas e gratuitas, aumenta consideravelmente o número de desenvolvedores para esta plataforma (Há..., 2012).

2.4 ALTERNATIVAS AS SMART TVS

O projeto Ginga baseado em software livre, propõe uma forma de desenvolver aplicativos que possam interagir com aparelhos de TV, desenvolvido pelo Pontífice Universidade Católica do Rio de Janeiro (PUC-RJ), com objetivo de criar um Middleware, que é uma camada de softwares que fica entre o código da aplicação e da infraestrutura das TVs que oferecem serviços de IPTV no sistema de TV nipo-brasileira Digital Terrestre (ISDB-TB), esta linguagem permitem o desenvolvimento rápido e fácil de aplicações (Ginga). Como o crescimento tecnológico de televisores se deu de uma forma relativamente rápida, muitas pessoas que haviam adquirido seu aparelho a pouco tempo, mas que ainda não

faziam parte deste novo grupo de televisores inteligentes, se viram de certa forma frustradas, por ter investido um valor não tão baixo em um equipamento que em pouco tempo já se tornou ultrapassado. Pensando neste mercado, algumas empresas voltaram seus olhos para tecnologias que pudessem agregar as funcionalidades destes novos equipamentos aos televisores comuns.

Com objetivo de transformar um televisor comum em uma *Smart TV*, o EQUISO é um dispositivo semelhante a um *pen drive*. Ao conectá-lo em uma TV através de uma entrada HDMI, adiciona a esta as funções de um *tablet* com sistema *Android*. Além destas funcionalidades o dispositivo ainda vem acompanhado com um controle que incorpora um teclado QWERTY. Desta forma, temos em uma televisão as mesmas funcionalidades de um *tablet*, podendo rodar aplicativos de jogos, redes sociais entre as diversas variedades existentes. O dispositivo vem com o *Android 4.0 (Ice Cream Sandwich)* e equipado com um processador de 1 Ghz, 512 MB de memória DDR3 e ainda um processador gráfico interno para que possa rodar aplicações 3D (Equiso Smart TV, 2012).

A possibilidade de criar em uma TV um ambiente semelhante ao de um computador é realmente fascinante, e podemos observar isso na busca por outras formas de transformar uma TV em uma *Smart TV*, sem que para isso tenha que se gastar muito dinheiro.

A opção oferecida pela empresa Apple é um dispositivo que conectado a um televisor também oferece acesso a serviços on-lines como YouTube, Vimeo e Flickr. Este dispositivo chamado de Apple TV, é bem compacto com 23 milímetros de altura, 98 milímetros de largura e 98 milímetros de espessura, este modelo é compatível com TVs de alta definição com HDMI e 1080p ou 720p a 60/50Hz, incluindo modelos populares dos seguintes fabricantes: Hitachi, JVC, LG, Mitsubishi, NEC, Panasonic, Philips, Pioneer, Samsung, Sharp, Toshiba, Vizio e Westinghouse (Referências..., 2013).

No dia 01/02/2013 (um de fevereiro de dois mil e treze) sites como Shopping Uol (<http://shopping.uol.com.br>) e Submarino (<http://www.submarino.com.br>) cotavam preço médio de R\$ 350,00 (Trezentos e cinquenta reais) pelo aparelho.

A Google TV também faz parte deste novo mercado mundial, oferecendo serviços semelhantes ao dos seu concorrentes com opção de conectar sua TV através de um dispositivo externo chamado de *Buddy Box* ou mesmo

utilizando televisores que já integram este dispositivo internamente, em 2013 a Google espera apresentar a nova plataforma da TV Google na International Consumer Electronics Show (CES) , em Las Vegas. (Kothari & Mickey, 2013)

2.5 SMART TVS E SUAS FORMAS DE INTERAÇÃO

Uma das principais variações das *Smart TVs* em relação aos equipamentos tradicionais, e que ainda não se tornou um consenso, sendo aplicado de diversas maneiras por fabricantes diferentes, refere-se a forma de interagir com estes equipamentos. Cada empresa busca sempre inovar e utilizar o que há de mais tecnológico disponível para agregar valores ao seu equipamento, sendo assim muitas novidades surgiram neste quesito.

A utilização do controle como uma espécie de mouse é uma das características apresentadas por alguns dos aparelhos. A Panasonic, por exemplo, criou um tipo de mouse que se assemelha com o *touchpads* de um *notebook*, bastando que você deslize os dedos para controlar o cursor na TV, ao apertar o botão o cursor clica na área selecionada , ver Figura 2.



Figura 2 - Controle da *Smart TV Panasonic touchpads*.

Uma outra solução também apresentada pela Panasonic é um aplicativo que pode ser baixado em seu *Smartphone* ou *tablet*, tanto da marca Apple como *Android*, e utilizá-lo como controle da TV.

A LG por sua vez propôs como controle um dispositivo que se assemelha ao controle do videogame Wii, com o qual você aponta para a tela da TV

e o movimento que é realizado com o controle é reproduzido com o cursor na tela da TV, como mostra a Figura 3. Além deste recurso, atrás do controle é oferecido um teclado que facilita a entrada de textos para navegação na internet.



Figura 3 - Controle que simula Mouse na TV da LG

Uma outra forma de interação apresentada pela Samsung se assemelha em partes ao equipamento da LG. Ela utiliza leitura corporal, através de uma câmera instalada no topo da TV, que captura os movimentos da mão e os utiliza para movimentar o cursor. Quando a mão é fechada a TV interpreta como um comando para clicar no item selecionado. Com outros movimentos consegue-se trocar de canal ou aumentar o volume do aparelho. Além da leitura corporal outro recurso interessante da *Smart TV* da Samsung é o controle por comando de voz, ver Figura 4 . Por meio de palavras ou frases específicas, pode-se abrir aplicativos ou navegar na *Web*. (Se..., 2012).



Figura 4 - Formas de interação Smart TV Samsung

2.6 A TECNOLOGIA DLNA

A *Digital Living Network Alliance* (DLNA) estabeleceu um padrão para compartilhamento de mídia entre os dispositivos em sua rede doméstica. *Smart TVs* que possuem certificação DLNA pode reproduzir filmes, fotos e música do seu PC, *Smartphone*, e muitos outros dispositivos que se conectam a sua rede sem fio. (Kovach, 2010)

Desta forma é possível que você possa facilmente apresentar em sua TV imagens e filmes que estejam em seu computador ou *Smartphone*, desde que ambos estejam conectados a uma mesma rede. Assim, torna-se fácil, por exemplo, ao tempo em que toda a família está reunida na frente da TV, exibir fotos ou vídeos que você armazenou durante o dia em outros dispositivos, sem que você precise transferi-los todos para o computador e que seus familiares precisem se apertar na frente do computador.

Apesar de novas tecnologias aproximarem ainda mais características dos televisores com os computadores existem entre estes dois equipamentos diversas diferenças. Os computadores são geralmente utilizados com foco em tarefas específicas (editar um texto, realizar compras, conversar com um amigo, jogar), tarefas estas que exigem atividades constantes dos usuários enquanto as TVs são equipamentos que oferecem uma interação um tanto quanto passiva, já que o telespectador assiste a um programa sem realizar grande interação com o equipamento além de trocar de canal e regular o volume.

Isto deixa clara a diferença entre a qualidade e quantidade de recursos (como tempo, atenção, memória, etc) que um usuário ou um telespectador estão dispostos a investir com cada equipamento.

A TV já existe há muito tempo e é praticamente unanimidade no Brasil. Segundo a pesquisa do CETIC.br, 98% dos domicílios brasileiros possuem pelo menos uma televisão (CETIC.br, 2011). O período em que a TV já existe na maioria dos lares fez com que se cria-se uma expectativa em relação ao seu uso, pois está muito bem definido para os espectadores buscar na televisão, diversão, entretenimento e informação, sem ter que realizar muitas tarefas para isso (Cybis, Betiol, & Faust, 2010).

As mudanças que os televisores sofreram estão dando oportunidades para que uma nova forma de assistir TV aconteça, as diversas formas em que um usuário pode interagir com sua TV exige que pensemos em novos meios para motiva-lo a realizar determinadas ações com sua TV. A possibilidade de integrar informações de meios externos, como aparelhos celulares e computadores a televisão permite tornar a televisão novamente o ponto central dos lares.

Para que isso aconteça todos estes novos recursos devem ser utilizados de forma moderada na construção de aplicativos que tornem simples e agradáveis sua utilização, e não afastem os usuários de tarefas que antes lhe eram prazerosas.

3 ARQUITETURA DAS SMART TVS

3.1 ARQUITETURA

Uma das formas mais prováveis de se criar uma interação entre transmissor e telespectador durante uma transmissão ao vivo é utilizar a transmissão via cabo ou antenas para enviar conteúdos de transmissão como vídeos, e utilizar a internet como um canal de retorno das informações do telespectador.



Figura 5 - Imagem de transmissão de TV

Isto por que o envio de vídeos via *streaming* diretamente de sua conexão de internet, poderia tornar a programação um tanto quanto incomoda, levando em consideração a estrutura que se tem hoje em nosso país para utilização da internet, poderíamos ter que aguardar enquanto um vídeo é carregado. Além do grande número de requisições que receberia um servidor que transmitiria este vídeo. Este seria, por exemplo, o cenário da transmissão de um jogo da final da copa do mundo. Utilizando a própria transmissão oferecida pela emissora, o aplicativo ficaria responsável apenas por apresentar aos usuários informações sobre os canais disponíveis, além de poder acessar os canais diretamente através do aplicativo, auxiliando o usuário na busca dos canais e podendo ainda oferecer conteúdo personalizado para cada canal, como vídeos, notícias e produtos.

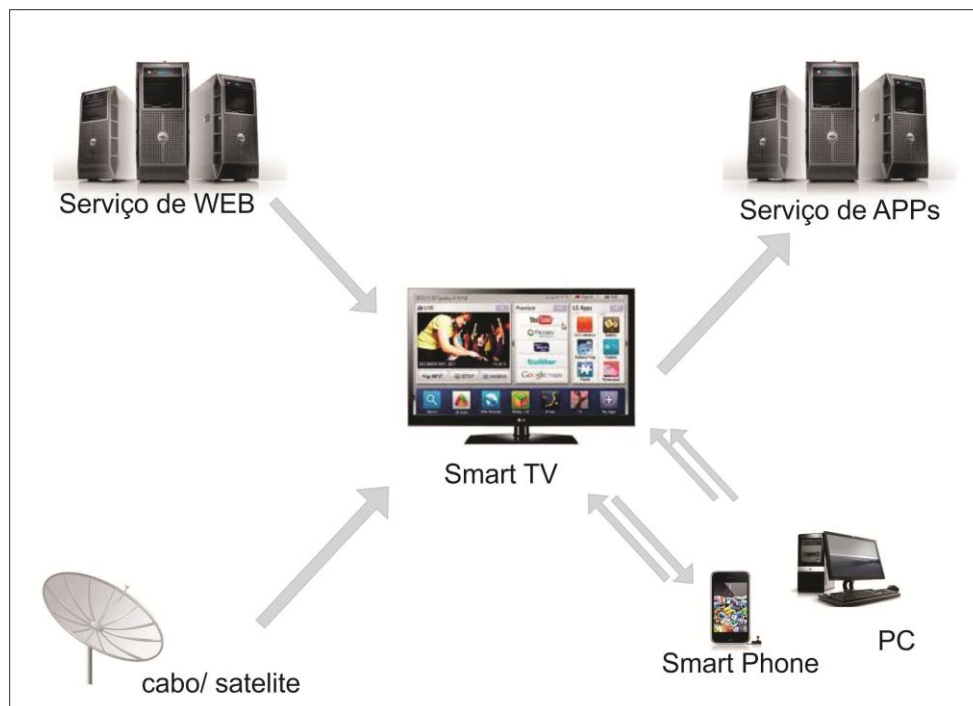


Figura 6 - Transmissão da Smart TV

No sistema de transmissão das *Smart TVs*, o conteúdo das emissoras é transmitido normalmente pelo padrão existente via cabo ou satélite, com a diferença de receber conteúdos também através dos aplicativos específicos que podem ser baixados na TV ou ainda navegando pela internet através de um navegador que já vem disponível na TV, através do sistema DLNA, pode-se ainda conectar outros dispositivos que estejam na mesma rede para fornecer conteúdos a sua TV como, um celular, Computador ou Câmeras.

A tabela abaixo apresenta alguns dados técnicos sobre a tecnologia aplicada a *Smart TVs* da Samsung, onde é apresentadas informações sobre sistema operacional da *Smart TV* e resoluções suportadas por esta.

Tabela 1 - Especificação Smart TV Samsung

Feature	Descrição
OS	Linux 2.6
Resolução gráfica	App: 960 x 540 32 bpp Smart Hub:1280 x 720 32 bpp Video: 1920 x 1080 32 bpp

Tabela 2 - Tecnologias suportadas pela Smart TVs Samsung

TYPE		FEATURE	TV/AVTV/AV	SDK 3.5.2
App Engine		HTML	HTML 5	HTML 5
		DOM	DOM 3	DOM 3
		CSS	CSS 3	CSS 3
		Javascript	SquirrelFish	SquirrelFish
Flash	Browser Plug-in	SWF	Flash 10.1 / ActionScript 3.0	Flash 10.1 / ActionScript 3.0
	Standalone	SWF	Flash 10.1 / ActionScript 3.0	Flash 10.1 / ActionScript 3.0
	AIR	AIR	AIR for TV 2.5.1 (TV D6000 or higher, BD D6700 or higher only)	Not Supported
	Streaming	Streaming	RTMP/RTMPe	RTMP/RTMPe
DRM		WMDRM 10 PD	Replaced (PlayReady)	Not Supported
		PlayReady	Supported	Not Supported
		Widevine	Supported	Not Supported
VOD	Streaming	HTTP	Supported	Supported
		HTTPS	Supported	Supported
		MMS	Supported	Supported
		RTP/RTSP	Supported	Supported
	Adaptive Streaming	Mpeg-Dash (xml metadata)	Supported - OIPF Rel2/MPEG2 TS only/AES-128	Supported
		HLS (m3u8 metadata)	Supported - HLS v3/MPEG2 TS only/AES-128	Supported
		HLS audio	Not Supported	Not Supported

3.2 FERRAMENTAS E APIS

A plataforma de desenvolvimento oferecida pela LG baseia-se em tecnologia livre e código aberto. Ela pode ser baixada em <http://opensource.lge.com> ou ainda pode ser solicitado o envio do conteúdo em CD. Esta plataforma contém a ferramenta Eclipse, mostrada na Figura 7, conhecida por muitos desenvolvedores que programação em C, C++, Java, PHP, etc.

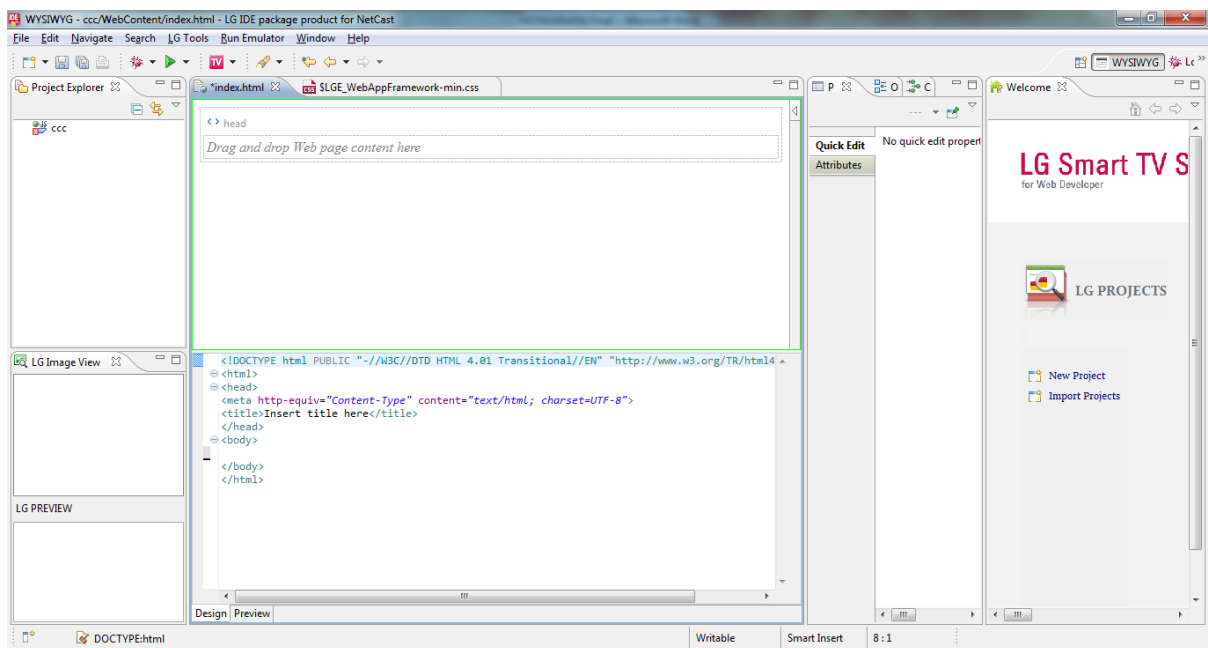


Figura 7 - Interface do SDK LG

Ao instalar o SDK juntamente é instalado um emulador para testar sua aplicação. Durante um projeto é possível determinar em qual versão dos equipamentos da fabricante deseja testar. Nos testes realizados para o desenvolvimento deste trabalho, foram aplicados a versão 2011, já que o emulador da versão 2012, não apresentou um funcionamento adequado, como não é o foco deste trabalho, não nos ativemos a realizações de teste para solucionar ou demonstrar o motivo do problema. O emulador apresenta uma tela de TV onde é carregada a aplicação que esta sendo testada, bem como um controle remoto, que permite interagir com a TV. No emulador o cursor do mouse é reconhecida como o cursor que é controlado pelo controle padrão da Smart TV da LG.

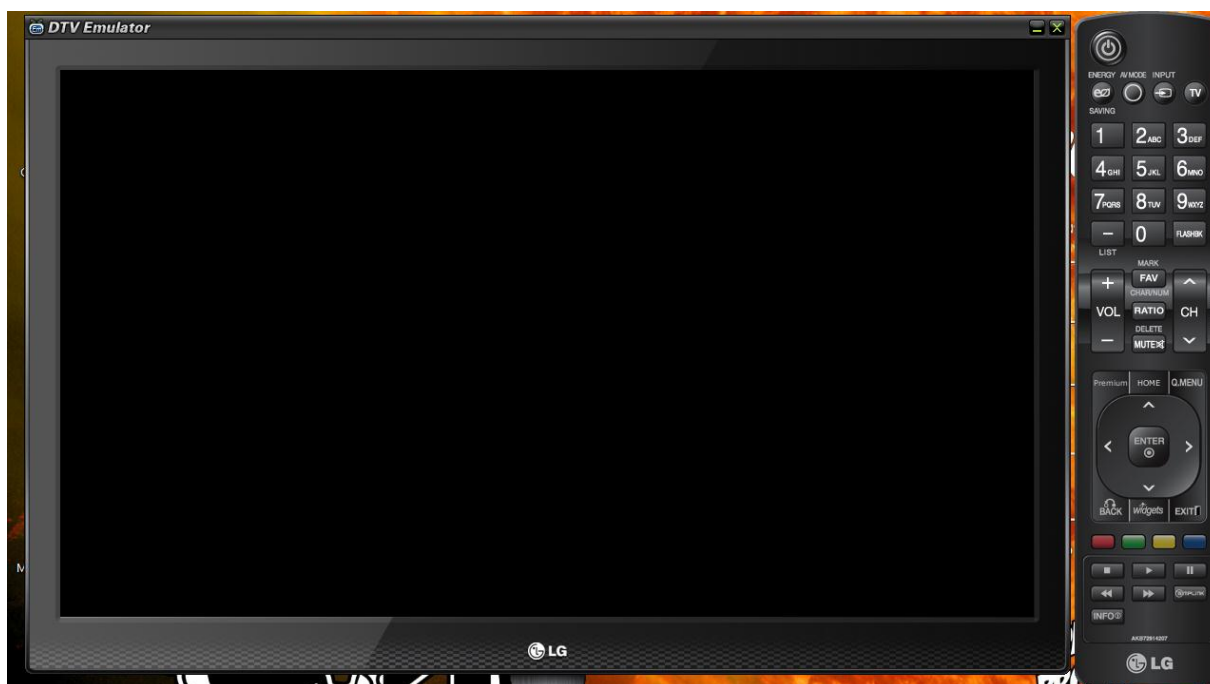


Figura 8 - Emulador com controle da LG

As *Smart TVs* oferecem através dos aplicativos acesso a recursos que podem ser realizados diretamente pelo controle como: controlar áudio, canais, etc. Cada uma das empresas pode implementar esta forma de acesso de diversas formas. Tanto a plataforma da LG como da Samsung, oferecem APIs para controlar seus equipamentos. Em seus fóruns de discussões podem ser encontrados tutoriais e informações sobre a implementação destes recursos.

A plataforma de desenvolvimento da Samsung, possui um aspecto mais personalizado, dando ao desenvolvedor, mais recursos de interfaces de desenvolvimento, com funcionalidade prontas,. Ela apresenta vários componentes prontos que podem ser inserido no aplicativo apenas clicando e arrastando com o mouse. São elementos como botões, caixas de texto, imagens, rótulos, elementos de vídeo, como mostra a Figura 9 . Juntamente com a ferramenta de desenvolvimento estão disponíveis emuladores com a versão 2010, 2011, e 2012.

Figura 9 - Interface de desenvolvimento da Samsung

O emulador da Samsung, permite que você logue com sua conta Samsung e tenha informações sobre os seus aplicativos.

Além do emulador ainda é exibido um terminal mostrado na figura 10, que apresenta todos os processos que estão sendo executados na TV. A cada comando realizado é apresentado no terminal os eventos acionados, como apertar

uma tecla do controle, estes eventos fazem parte da API da TV, e podem ser consultados na documentação existentes no site do fabricante.

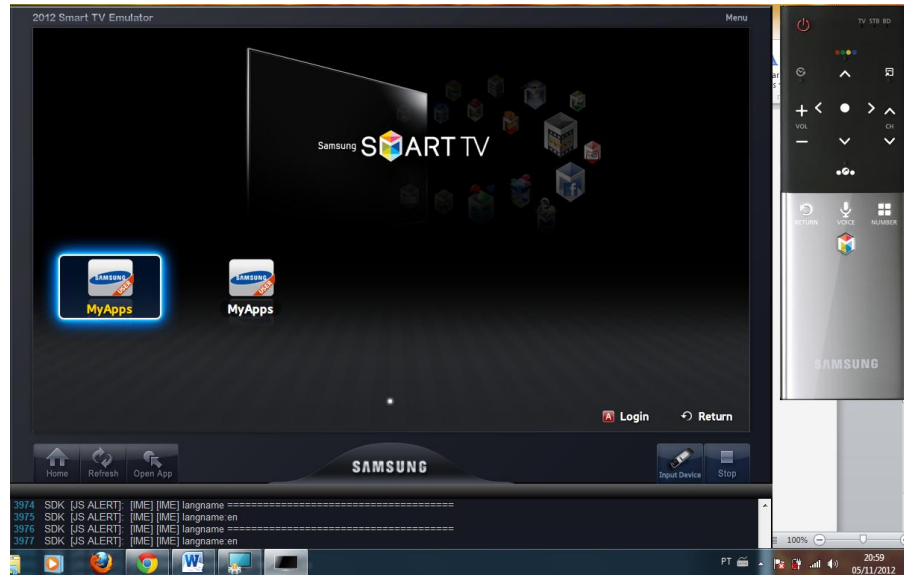


Figura 10 - Emulador 2012 da Samsung

Alguns exemplos da utilização da API em determinados eventos como, selecionar um canal ou altera-lo, estão disponíveis como exemplos que podem ser baixados para testar. O código a baixo demonstra como capturar informações do canal que esta sendo exibido através da função da API LG e em seguida alterar um canal utilizando a função *nextProgramInfoHandler*.

```
function currentProgramInfoHandler () {
    var currentProgram = broadcast.getCurrentProgram();
    jsLog.lgmethod('getCurrentProgram()');
    htmlStr1 += "Title: " + currentProgram.title
    + "<br>Start Time: " + currentProgram.startTime + "<br>End Time: " +
    currentProgram.endTime + "<br> Description: "
    + currentProgram.description + "<br> Event Id: "
    + currentProgram.eventId + " ";
    showPropResult ();
}
```

Figura 11 - Função da API LG para buscar informações de um programa exibido

```

function nextProgramInfoHandler () {
    var nextProgram = broadcast.getNextProgram();
    jsLog.lgmethod('getNextProgram()');
    htmlStr1 += "Title: " + nextProgram.title +
    "<br>Start Time: " + nextProgram.startTime +
    "<br>End Time: " + nextProgram.endTime +
    "<br> description: " + nextProgram.description +
    "<br> Event Id: " + nextProgram.eventId + " ";
    showPropResult();
}

```

Figura 12 - Função da API LG apresentar informações do canal

Outra função que seria essencial para construção deste trabalho seria a possibilidade de executar dentro do aplicativo a transmissão da programação que esteja passando na TV. Buscamos então na documentação por recursos da API que permitissem este acesso aos dados da TV, e para realização desta tarefa a LG oferece um recurso de *BroadCast* que permite criar um objeto que pode manipular e oferecer informações da transmissão da TV. Abaixo é apresentado o código que é responsável por alternar o canal que esta sendo transmitido, utilizando o método *channelUpHandler* e *channelDownHandler*.

```

function channelUpHandler () {
    broadcast.channelUp();
    jsLog.lgmethod('channelUp()');
}

function channelDownHandler () {
    broadcast.channelDown();
    jsLog.lgmethod('channelDown()');
}

```

Figura 13 - Métodos responsáveis por alterar o canal de transmissão

A documentação da Smart TV da LG oferece ainda um modelo de código fonte completo para ser implementado, que possa apresentar todas as funções de controle de uma transmissão bem como interagir com ela, logo a seguir e exibido a Figura 14 que ilustra este exemplo de implementação oferecido pela LG, seu código de exemplo segue anexo a este trabalho.

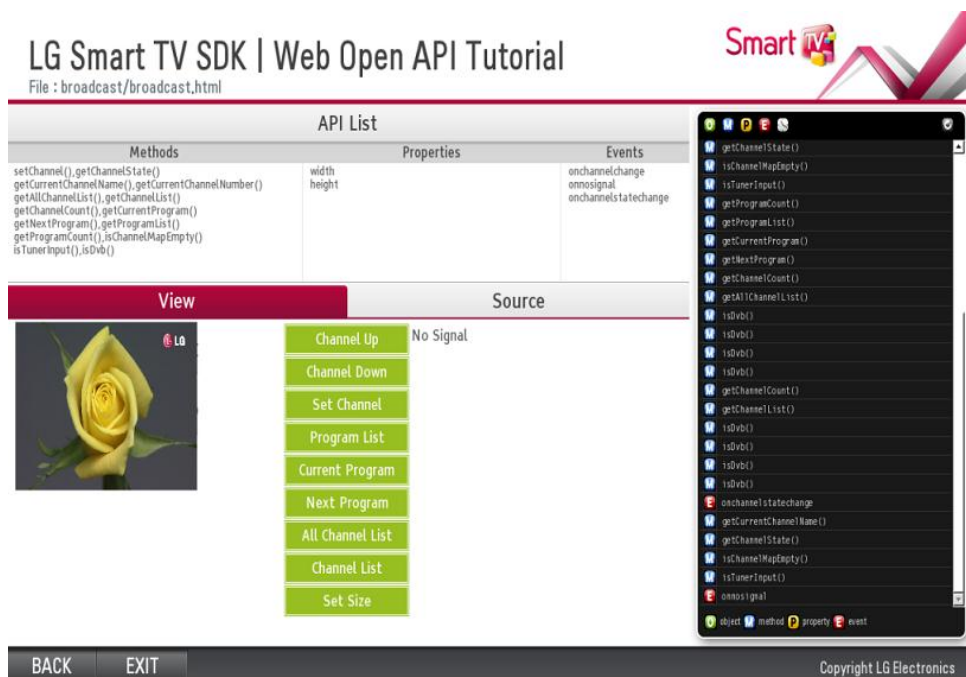


Figura 14 - Implementação do exemplo de BroadCast da LG

Para implementação deste exemplo diversos métodos da API são utilizados, abaixo é apresentado uma tabela com cada um destes métodos e suas funções.

Tabela 3 - Eventos do objeto BroadCast da Smart TV LG adaptado de Smart TV Dcumentation

API Class	Function Name	Descrição
	channelUp	Mudas de canal utilizando a tecla acima do controle remoto.
	channelDown	Mudas de canal utilizando a tecla abaixo do controle remoto.
	getCurrentChannelName	Retorna o nome do canal atual.
	getCurrentProgram	Usado para ver a informação do programa transmitido no canal atual.
	getNextProgram	Usado para visualizar informações do programa, que estão vindo após o programa transmitido no canal atual.
Method	getCurrentChannelNumber	Usado para ver o número de identificação do programa ID do canal atual.
	getProgramList(startTime, endTime)	Obter informações de data do programa atual
	getProgramCount	Usado para obter o número de programas para a data especificada do canal atual.
	setChannel (signalType, physicalNum, logicalNum)	Usado para mover diretamente para o canal especificado.
	getChannelList(serviceType,signalType)	Usado para obter as informações de canais pesquisáveis da TV atual. Com base no índice de, no máximo 10 valores de resultados são retornados.
	getChannelCount(serviceType,inputType)	Usado para obter o número de canais pesquisáveis da TV atual.

	getChannelState	Usado para obter o status de sinal de TV para o programa atual.
	isChannelMapEmpty	Utilizado para determinar se existe um mapa do canal de televisão actual.
	isTunerInput	Usado para determinar se o sinal de entrada é de TV.
	isDvb	Usado para determinar se o sinal de TV é compatível com padrão DVB ou ATSC.
Property	Width	Retorna a largura o objeto de transmissão
	Height	Retorna a altura o objeto de transmissão
Event	Onchannelchange	Ocorre quando um canal é alterado.
	Onnosignal	Ocorre quando não a canal sintonizado
	Onchannelstatechange	Este evento ocorre quando o estado do sinal de TV é alterada.

Além de Métodos utilizados para controlar a transmissão da programação, também é oferecido métodos para interação com a TV, como reconhecimento de voz que pode ser acionado através da implementação dos métodos seguintes.

Tabela 4 - Eventos de reconhecimento de voz da Smart TV LG

API Class	Function Name	Description
Method	startRecognition	Chamar UI nativa da função de reconhecimento de voz e receber o resultado como um evento.
Property	isInitialized	Determinar se a função de reconhecimento de voz é inicializado e retorna o valor booleano.
	isEnabled	Determine se o MRCU está emparelhado (incluindo seu tipo) e se a função de reconhecimento de voz está ativada, e retorna o valor booleano.
	Dictation	Determinar se a função de reconhecimento de voz está em modo de ditado e retorna o valor como string (on / off). Se o valor de retorno é "off", a função está em modo de palavra-chave. Estes métodos também podem ser utilizados para selecionar o modo
Event	onrecognizevoice	Receber o resultado de reconhecimento de voz da TV.
	onbuttonenable	Ativar ou desativar o botão de reconhecimento de voz

São diversos os recursos oferecidos pela *Smart TVs* e as empresas fabricantes destes equipamentos vêm buscando auxiliar os desenvolvedores nesta tarefa, oferecendo documentação e criando um espaço de discussão para troca de experiência entre os mesmos. Estes espaços porém ainda estão em pouco uso. No Brasil, por exemplo, é difícil encontrar fóruns de discussões sobre o tema, o que demonstra que os desenvolvedores ainda não viram todo o potencial que esta nova tecnologia pode oferecer.

4 ESCOPO DO PROJETO

Os requisitos deste trabalho englobam o planejamento de um protótipo de aplicativo para *Smart TV*, que aborda transmissões esportivas demonstrando quais as possíveis formas de se interagir durante a realização destes eventos. Para tanto, elaborou-se o conjunto de requisitos abaixo.

4.1 REQUISITOS DO SISTEMA

O aplicativo deve oferecer um calendário com as datas e horários das partidas a serem realizadas, permitindo ao usuário acessar através do aplicativo qualquer partida que esteja em andamento, bem como buscar informações sobre a partida, como escalação, estatísticas, etc.

O telespectador poderia ainda dar um palpite para partida podendo se premiado com pontos pelos acertos, e estes poderiam gerar algum benefício posterior.

O aplicativo deve oferecer ao usuário informações das partidas que estão ocorrendo, e das que ocorreram, permitindo o acompanhamento de lances polêmicos e jogadas realizadas.

A exibição de uma tabela de pontuação atualizada de forma dinâmica com resultados de partidas em andamento, bem como a simulação de resultados das partidas.

Aproveitando a interação gerada pelas redes sociais, pode-se dizer que a integração com algum tipo de rede social e o aplicativo é considerável, tendo que o ato de assistir uma partida de futebol, por exemplo, é algo que reúne pessoas, e sempre gera debates e discussões que podem ser realizadas agora sem que estas pessoas estejam próximas fisicamente, mas sim virtualmente, compartilhando opiniões sobre a partida.

Estes recursos ao serem implementados no aplicativo permitiriam ao telespectador interagir com a programação possibilitando ainda a realização de outras tarefas, como comprar um produto.

Com a aplicação rodando em um servidor *Web*, o qual contém informações sobre a transmissão que está em andamento, é possível definir propagandas específicas para cada tipo de transmissão. Isto pode ser realizado direcionando a propaganda aos interesses do telespectador que pode acessar informações completas na aplicação se desejar. Esta seria uma forma interessante de se realizar *merchandising* em programas esportivos. Ao acessar o aplicativo pela primeira vez o usuário preenche informações por exemplo sobre o time que torce, assim durante uma partida não aconteceria de lhe ser direcionada uma propaganda oferecendo a camisa de um time rival, ou desconto em ingresso para a partida de outro time.

Se a emissora de TV disponibilizar mais de uma câmera para realizar a transmissão de uma partida e disponibiliza-la em outro canal, o aplicativo pode ficar incumbido de gerenciar esta informação dando ao telespectador a possibilidade de escolher o ângulo que deseja assistir a partida.

Durante uma partida o telespectador pode querer acompanhar a transmissão de outra partida que esteja em andamento, o aplicativo oferece a possibilidade de acionar este recurso pelo usuário e não somente pela emissora de TV, assim o aplicativo dividiria a tela em duas áreas separadas para cada transmissão.

Para um melhor entendimento de todas as funcionalidades que o sistema deve oferecer e também para representar o comportamento de entrada e saída de dados deste, são listadas a baixo os requisitos funcionais para o desenvolvimento deste aplicativo.

4.1.1 Requisitos Funcionais

Os requisitos funcionais do sistema referem-se diretamente as funcionalidade que o aplicativo deve possuir para atender as necessidade dos usuários, segundo (Sommerville, 2007) os requisitos funcionais são as funcionalidades de cada exceção do sistema, os comportamentos, suas ações para cada entrada, ele depende do tipo de software que esta sendo desenvolvido bem como do usuários que utilizaram este software.

Tabela 5 – Requisitos funcionais do sistema

RF01	O usuário deve se identificar para acessar o aplicativo, sem seu primeiro acesso um breve cadastro deve ser realizado.
RF02	A área principal do aplicativo deve dar acesso a todos recursos de forma rápida e simples.
RF03	Durante as transmissões o aplicativo deve apresentar um menu lateral para acesso a funcionalidades relacionadas a programação
RF04	O sistema deve disponibilizar uma galeria de vídeo para o usuário
RF05	Durante a reprodução de um vídeo o sistema deve apresentar uma lista de vídeos relacionados
RF06	O sistema deve apresentar informações sobre partida como, tabela de classificação, e calendário de jogos
RF07	O sistema deve apresentar ao usuário opções de canais com suas descrições.
RF08	Durante a transmissão de uma partida o sistema deve possibilitar ao usuário acessar recursos, como comentários, câmeras, enquetes, classificação e informações da partida.
RF09	O sistema deve permitir ao usuário comentar vídeos.

4.1.2 Requisitos não funcionais

Os requisitos não funcionais do sistema referem-se diretamente a qualidade do sistema, ele pode ser dividido em varias partes como, segurança, usabilidade e desempenho, comunicabilidade. Alguns requisitos foram definidos para o bom funcionamento deste aplicativo, são eles.

Tabela 6 – Requisitos não funcionais do sistema

RNF01	O usuário deve se identificar para acessar o aplicativo, em seu primeiro acesso, e utilizar a escolha de usuário e senha quando houver mais usuários cadastrados no mesmo equipamento.
-------	--

RNF02	Os acessos aos recursos devem ser de forma fácil e compreensível, para evitar que o usuário tenha dificuldade ao procurar algum recurso.
RNF03	As ações do usuário devem produzir <i>feedback</i> , para que este saiba o que está acontecendo no sistema.
RNF04	Consistência, manter a padronização de elementos como botões e ícones para facilitar o reconhecimento pelo usuário.
RNF05	Criar interfaces sem poluição visual, apresentando somente as informações que o usuário necessita.

4.2 COLETA DE DADOS

Para realização deste trabalho foi imprescindível a realização de uma coleta de dados que permitisse destacar os requisitos necessários para construção do aplicativo aqui proposto. Para esta realizamos entrevistas com pessoas na faixa etária entre 15 e 40 anos que possuíssem conhecimento básico na utilização de Internet, e que possuíssem interesse em esportes, especificamente em futebol.

As entrevistas ocorreram informalmente, onde o entrevistado foi solicitado sempre a expressar suas sugestões, ideias e vontades de forma livre, sem que este se limitasse a questionários com opções pré-estabelecidas, dando assim a possibilidade de surgimento a funcionalidade que não havia ao menos sido cogitada. Com base nestes dados e nas tecnologias disponíveis foram definidos os requisitos do sistema que visa atender as necessidades expressadas por estas pessoas.

4.3 CENÁRIOS

Para melhor compreensão das funcionalidades do aplicativo proposto e como este deve se comportar em relação às ações do usuário foram criados diversos cenários que visam ilustrar de forma objetiva as funcionalidades do sistema, demonstrando a interface proposta deste projeto e apresentando uma

descrição textual que permita auxiliar na compreensão das tarefas que estão sendo realizadas.

4.3.1 Cenário 1 – acessando o aplicativo

No primeiro cenário ilustrado na Figura 15, um telespectador ao acessar o aplicativo pela primeira vez, deve oferecer algumas informações para realização de seu cadastro. Mais de uma pessoa pode utilizar o mesmo aplicativo, por isso será necessário utilizar contas de usuários para manter as configurações individuais de cada usuário. Por medida de segurança todas estas contas devem ser protegidas por senha.



Figura 15 – Acesso ao aplicativo

4.3.2 Cenário 2 – área de trabalho do aplicativo

No segundo cenário ilustrado na Figura 16, quando um usuário acessa o aplicativo, após se identificar, a área de trabalho do aplicativo deve ser carregada, esta conterá informações como notícias e vídeos. Esta área de trabalho deverá conter uma barra de acesso rápido às funcionalidades do aplicativo identificadas por ilustrações e texto que sejam condizentes com suas funcionalidades. Deverá conter uma área específica para enviar notificações para o usuário. Caso o usuário deseje poderá personalizar as informações que são exibidas na página inicial, por exemplo, especificando o clube para o qual torce e definindo que só quer receber informações deste.



Figura 16 - Tela da área de trabalho do aplicativo

4.3.3 Cenário 3 – lista de canais

No terceiro cenário ilustrado na Figura 17, quando o usuário acessar a lista de canais o aplicativo deve apresentar todos os canais disponíveis na parte lateral esquerda da janela, e ao selecionar um canal, na lateral direita será exibido informações sobre este. Se o canal for um programa de TV, apresentará dados deste programa como, nome, horário de início e término, apresentadores, etc. Se o canal selecionado se tratar da transmissão de uma partida de futebol serão apresentado dados referentes a esta partida, como clubes, local, campeonato, rodada. Além destas informações poderão ser exibidos dados referentes aos recursos utilizados nesta partida, como opção de narração, opção de câmera, promoções, etc.



Figura 17 – Lista de canais

4.3.4 Cenário 4 – área de vídeos do aplicativo

No quarto cenário ao acessar a área de vídeos o usuário poderá optar por visualizar vídeos específicos de seu clube ou geral. Em ambas as opções este terá a sua disposição o recurso que permite ao usuário visualizar vídeos relacionados, bem como comentar um vídeo, podendo ser aplicado aqui APIs como a do Facebook, como ilustrado na Figura 18, . O usuário também pode optar por visualizar o vídeo em tela cheia.



Figura 18 - área de vídeos

4.3.5 Cenário 5 – informações sobre campeonatos e clubes

No quinto cenário o aplicativo deverá fornecer ao usuário recursos que lhe permitam escolher um determinado campeonato e com base nestes dados

apresentar informações sobre partidas realizadas, com escalação, faltas, cartões, posicionamento etc. também deverá apresentar um calendário com os próximos jogos, e uma tabela de classificação do campeonato, e simulação de resultados.

4.4 RECURSOS DE VIDEOS E TRANSMISSÕES

Como o principal objetivo do aplicativo proposto é referente a transmissão e reprodução de vídeos optou-se por definir melhor as funcionalidades envolvidas nesta tarefa, desta forma será descrito como se dará a realização das seguintes tarefas:

- Comentar vídeo, ou transmissão;
- Escolher câmera;
- Escolher narrador;
- Acessar informações;
- Enquete;
- Promoções;
- Previsão de resultado;
- Replay;

4.4.1 Comentar vídeo ou transmissão

O comentário de vídeo ou transmissão ilustrados na figura 19, irá permitir que o usuário durante uma transmissão acesse um menu lateral do aplicativo e um painel se abra ao lado da programação para que este possa comentar o programa ou jogo que está sendo transmitido, o foco deste recurso é a aplicação da API do Facebook para permitir que usuário possa compartilhar sua opinião com seus amigos durante uma transmissão. O mesmo recurso estará disponível ao acessar um vídeo da galeria de vídeos ou o replay durante uma partida



Figura 19 - Comentários em vídeos

4.4.2 Câmera

O recurso de câmera está intrinsecamente ligado à emissora que está realizando a transmissão pois para que este recurso possa ser realizado, a emissora deve fornecer dois canais ou mais, cada canal irá oferecer um ângulo diferente de visualização, desta forma e responsabilidade do aplicativo apresentar ao usuário os canais disponíveis com os diferentes ângulos da câmera como mostra a Figura 20.

Apesar de não ser o foco deste trabalho, durante o levantamento de dados sobre as tecnologias utilizadas pelas TVs, pôde-se observar algumas características do novo padrão de TV digital que podem solucionar este problema já que este novo padrão permite a transição de uma maior quantidade de informações, e pode ser realizado em trabalhos futuros.



Figura 20 - Escolha De Câmera

4.4.3 Narração

O recurso de narração é um dos recursos mais abordados durante a coleta de dados, a possibilidade de se escolher a narração da partida é muito interessante porém depende de diversos fatores, a emissora deve disponibilizar narrações diferentes da mesma forma que esta disponibilizaria canais diferentes para transmissão do vídeo. Uma outra possibilidade seria realizar a transmissão do áudio através de *streaming* diretamente de um servidor, porém este poderia apresentar problemas para que fosse sincronizado com a transmissão. Este item também poderia ser estudado com base nas funcionalidade e recurso oferecidos pela TV digital. Ao acessar este recurso quando disponível para uma transmissão o áudio da partida seria substituído pelo áudio escolhido como mostra a Figura 21.



Figura 21 - seleção de áudio

4.4.4 Informação da partida

Ao acessar o recurso de informação durante uma partida como mostrado na Figura 22, o usuário teria a sua disposição dados sobre a partida, como estatísticas de posse de bola, faltas, escanteios, cartões, escalação do time, local da partida, campeonato, resultados de outros confrontos, e todos estes dados estariam disponíveis durante o decorrer da partida sendo atualizados frequentemente, estes dados são arquivados e podem ser acessados posteriormente pelos usuários através do aplicativo pela opção de campeonatos disponíveis na tela principal do aplicativo, também estarão disponíveis recursos para simular resultados.



Figura 22 - Estatísticas da partida

4.4.5 Enquetes

Este recurso pode ser implementado no aplicativo para realizar enquetes principalmente durante as partidas, os resultados são apresentados dinamicamente conforme os usuários vão respondendo as mesmas. Este recurso pode ser aplicado de diversas formas uma delas é a participação do usuário em questionamento sobre um jogo ou um tema específico.

4.4.6 Promoções e vendas

A *Smart TVs* abrem a possibilidade para uma nova forma de *e-commerce*, pois com os novos recursos poderá ser apresentado ao usuário produtos se este o

interessar basta dar um click para ser redirecionado para realizar a compra do produto, este recurso pode ser implementado até mesmo para que este possa realizar a compra pelo aplicativo bastando adicionar a este informações necessárias para realizar a compra, como e feito por exemplo com aplicativos *Apple Store*. Com base em informações da transmissão que o usuário esta assistindo é possível direcionar produtos que sejam interessantes para eles, o que proporcionaria uma possibilidade maior de conversões e vendas.

Este recurso pode ser implementado através de uma área no aplicativo para que este efetue a compra ali mesmo, ou apenas seja direcionado para página de venda do produto, permitindo a venda de anúncios dentro do aplicativo.

4.4.7 Previsão de resultados

Durante a transmissão de uma partida por um tempo predeterminado o telespectador pode fazer previsões de resultados, estas previsões quando concretizadas podem ser convertidas em pontos que podem ser trocado por brindes como ingresso, camisetas, etc. Além disso pode ser criado ainda um Rack com os usuários sendo classificados por seus acertos. Isto incentivaria a competitividade e participação dos usuários durante as partidas.

4.4.8 Replay

Este recurso oferece ao usuário durante uma partida a possibilidade de rever lances durante a execução jogo, na forma proposta, estes lances serão disponibilizados pela própria emissora em um servidor, ficando a cargo do aplicativo gerenciar o acesso a estes para o usuário, oferecendo a este os vídeos e lances da partida que esta acompanhando como ostra a Figura 23.



Figura 23 - Tela do replay da partida

5 CONCLUSÃO

Ao termino deste trabalho podemos afirmar que as novas possibilidades oferecidas pelas *Smart TV* são muitas, seja através de aplicativos, que existem e podem ainda ser desenvolvidos, quanto pelas diversas formas de interação que elas oferecem.

Este trabalho buscou expor as possibilidades oferecidas por estes novos equipamentos e não servir como um manual para construção de um aplicativo, sendo assim podemos afirmar que este novo segmento de desenvolvimento de softwares tem grandes possibilidades de ser um mercado de sucesso tão importante como os de aplicativos para celulares que já é uma realidade atual.

As informações sobre estes equipamentos ainda são limitadas, as referências são vagas o que dificulta em parte a realização de pesquisa sobre elas. Apesar dos fabricantes de *Smart TVs* oferecerem ferramentas para desenvolvimento e testes, o trabalho proposto não poderia realizar a construção de uma aplicação nas dimensões propostas sem que fosse necessária a utilização de um equipamento para testes, já que este trabalho está diretamente relacionado a interação com a transmissão do equipamento, sendo assim outros resultados não puderam ser apresentados.

Mesmo assim pode-se considerar que o trabalho aqui realizado apresentou resultados satisfatórios, pois permitiu explorar um tema ainda pouco abordado, e poderá ser utilizado para realização de outros trabalhos que podem buscar explorar novas possibilidades para realização de alguns dos problemas enfrentados aqui, novas tecnologias e novos recursos estão sendo implementados pelos fabricantes de *Smart TVs*, o desenvolvimento de outros trabalhos que apliquem este tema vinculado a temas das *TVs* digital podem apresentar resultados interessantes na solução de problemas apresentados aqui, isso porque as duas tecnologias podem oferecer outros meios de transmissão.

O mercado brasileiro está caminhando nesta fase de desenvolvimento de aplicativos para *Smart TVs*, e aparentemente as pessoas ainda não descobriram o que realmente tem disponível em seus novos aparelhos de TV. Nem as empresas se

ativeram para este novo canal de contato direto com seus consumidores, a oportunidade de se destacar e sair na frente é agora.

REFERÊNCIAS

DIGITAL_NEWS. (06 de Setembro de 2012). Acesso em 26 de Setembro de 2012, disponível em Olhar Digital: http://olhardigital.uol.com.br/produtos/digital_news/noticias/telespectadores-de-paises-emergentes-usam-mais-tvs-conectadas,-diz-estudo

Developers. (2012). Acesso em 24 de Setembro de 2012, disponível em <http://www.smarttv-alliance.org>: <http://www.smarttv-alliance.org/Markets/Developers.aspx>

Equiso Smart TV. (28 de junho de 2012). Acesso em Setembro de 24 de 2012, disponível em kickStarter: <http://www.kickstarter.com/projects/2028163448/equiso-smart-tv-turn-any-tv-into-a-smart-tv>

Aplicativos... (2012). Navita. Acesso em 20 de Setembro de 2012, disponível em <http://www.navita.com.br>: <http://www.navita.com.br/portal/quem-somos/definicoes/aplicacoes-moveis.html>

CETIC.br. (2011). cetic.br. Acesso em 29 de fevereiro de 2013, disponível em cetic.br: <http://cetic.br/usuarios/tic/2011-total-brasil/rel-geral-00.htm>

Cybis, W., Betiol, A. H., & Faust, R. (2010). Ergonomia e Usabilidade Conhecimentos, Métodos e Aplicações. São Paulo: Novatec.

GFK. (31 de Agosto de 2012). Press Release . Acesso em 28 de Fevereiro de 2013, disponível em www.gfk.com: http://www.gfk.com/imperia/md/images/significant/press_release_smart_tv.pdf

Ginga. (s.d.). Sobreo Ginga. Acesso em 28 de Fevereiro de 2013, disponível em www.ginga.org.br: <http://www.ginga.org.br/pt-br/sobre>

Há... (22 de Julho de 2012). Descubra as vantagens de uma plataforma única de Smart TV. Acesso em 25 de Setembro de 2012, disponível em Olhar Digital: http://www.olhardigital.com.br/produtos/central_de_videos/ descubra-as-vantagens-de-uma-plataforma-unica-de-smart-tv

Ig, P. (22 de Junho de 2012). <http://tecnologia.ig.com.br>. Acesso em 24 de Setembro de 2012, disponível em <http://www.ig.com.br>: <http://tecnologia.ig.com.br/2012-06-22/ig-e-philips-confirmam-alianca-para-padronizar-aplicativos-para-tvs.html>

Kothari, S., & Mickey, K. (4 de Janeiro de 2013). Google TV. Acesso em 1 de Fevereiro de 2013, disponível em <http://googletv.blogspot.com.br>: <http://googletv.blogspot.com.br/search?updated-min=2013-01-01T00:00:00-08:00&updated-max=2014-01-01T00:00:00-08:00&max-results=1>

Kovach, S. (8 de Dezembro de 2010). what is a smart tv? Acesso em 5 de Agosto de 2012, disponível em Businessinsider: <http://www.businessinsider.com/what-is-a-smart-tv-2010-12>

Referências... (2013). Especificações técnicas. Acesso em 1 de Fevereiro de 2013, disponível em <http://www.apple.com>: <http://www.apple.com/br/appletv/specs.html>

samsungdforum. (s.d.). Acesso em 24 de setembro de 2012, disponível em <http://www.samsungdforum.com>: <http://www.samsungdforum.com/Devtools/Spec>

Se... (15 de Janeiro de 2012). As novidades do mundo das TVs Conectadas. Acesso em 24 de Setembro de 2012, disponível em Olhar Digital: http://olhardigital.uol.com.br/produtos/central_de_videos/ces-2012-quais-as-principais-novidades-em-tvs-conectadas

Sobre... (2012). Sobre aplicativos. Acesso em 24 de Agosto de 2012, disponível em <http://support.google.com>: <http://support.google.com/chrome/bin/answer.py?hl=pt-BR&answer=1050586>

Sommerville, L. (2007). Engenharia de Software (Vol. 8). São Paulo: PEARSON EDUCATION.

ANEXO A – Exemplo de código fonte do aplicativo de broadcast da LG

```

DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Broadcast Api Test</title>
<link rel="stylesheet" href="../css/style.css">
<script language="javascript" src="../js/common.js"></script>
<script language="javascript" src="../js/keycode.js"></script>
<script language="javascript" src="../js/menu.js"></script>
<script type="text/javascript" src="../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="stylesheet" href="../blackbirdjs/blackbird.css"
/>
<script>
    addEventHandler(window, "load", initPage);
    var htmlStr1 = "";
    var broadcast;
    var msgArea;
    //initialize page
    function initPage() {

        //save page as last visited page
        setLastVisitPage();

        //common initialize function
        commonInitialize();
        equestSourceCode();
        jsLog.initLG();

        broadcast = document.getElementById('broadcast');
        msgArea = document.getElementById('displayArea');

        checkNetCastVersion();

        //add onclick event handler
        addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
        addEventHandler(document.getElementById("btn_exit"), "click",
onClickHandler);
        addEventHandler(document.getElementById("channel_up"), "click",
channelUpHandler);
        addEventHandler(document.getElementById("channel_down"), "click",
channelDownHandler);

        jsLog.lgobject('application/x-netcast-broadcast');

        loadContent();
        showPropResult()
    }

    function processChannelChangeFunction(e) {
        if (e) {
            jsLog.lgevent('onchannelchange');
            loadContent();
            showPropResult();
        }
    }

    function processChStateChangeFunction(e) {
        if (e) {
            sLog.lgevent('onchannelstatechange');
            loadContent();
            showPropResult();
        }
    }

```

```

    }
}

function loadContent() {
    htmlStr1 = (broadcast.getCurrentChannelName() != undefined) ? "<font
style='color:#000000; font-weight:bold'>Current Channel Name:</font> " +
broadcast.getCurrentChannelName() : "";
    jsLog.lgmethod('getCurrentChannelName()');
    htmlStr1 += (broadcast.getChannelState() != undefined) ? "<br><font
style='color:#000000; font-weight:bold'>Channel State:</font> " +
broadcast.getChannelState() : "";
    jsLog.lgmethod('getChannelState()');
    htmlStr1 += (broadcast.getCurrentChannelName() != undefined) ? "<br><font
style='color:#000000; font-weight:bold'>Is ChannelMap Empty:</font> " +
broadcast.isChannelMapEmpty() : "";
    jsLog.lgmethod('isChannelMapEmpty()');
    htmlStr1 += (broadcast.getCurrentChannelName() != undefined) ? "<br><font
style='color:#000000; font-weight:bold'>Is Tuner Input:
</font> " + broadcast.isTunerInput() : "";
    jsLog.lgmethod('isTunerInput()');

    var currentChannel = broadcast.getCurrentChannelNumber();
    var signalType = currentChannel.signalType;
    var physicalNum = currentChannel.physicalNum;
    var logicalNum = currentChannel.logicalNum;
    var minorNum = currentChannel.minorNum;
    var majorNum = currentChannel.majorNum;
    var programNum = currentChannel.programNum;
    var sourceId = currentChannel.sourceId;
    var serviceType = currentChannel.serviceType;
    var tsId = currentChannel.tsId;
    var nwId = currentChannel.originalNetworkId;
    htmlStr1 += (currentChannel != undefined) ? "<br><font style='color:#000000;
font-weight:bold'>Current Channel Info:</font><br>1. Signal Type: " + signalType +
"<br>2. Physical Number: " + physicalNum + "<br>3. Logical Number: " + logicalNum +
"<br>4. Major Number: " + majorNum + "<br>5. Minor Number: " + minorNum + "<br>6.
Program Number: " + programNum + "<br>7. Souce ID: " + sourceId + "<br>8. Service
Type: " + serviceType + "<br>9. TS ID: " + tsId + "<br>10. Original Network ID: " +
nwId : "";

}

function showPropResult() {
    msgArea.innerHTML = htmlStr1;
    msgArea.style.overflow = "auto";
    htmlStr1 = "";
}

function processNoSignalFunction(e) {
    var htmlStr = "";
    if (e) {
        htmlStr += "No Signal";
        jsLog.lgevent('onnosignal');
    }
    msgArea.innerHTML = htmlStr;
}

//onUserInput function should be implemented
function onUserInput(userInput) {

    switch (userInput) {
        case VK_BACK:
            window.location.replace("../main_menu.html");
            break;

    }

}

function channelUpHandler() {

```

```

        broadcast.channelUp();
        jsLog.lgmethod('channelUp()');
    }

    function channelDownHandler() {

        broadcast.channelDown();
        jsLog.lgmethod('channelDown()');
    }

    function allChannelListHandler() {
        var channelList;
        var channelCount;
        var channelName = new Array();
        var signalType = new Array();
        var physicalNum = new Array();
        var majorNum = new Array();
        var minorNum = new Array();
        var logicalNum = new Array();
        var channelCount = broadcast.getChannelCount("tv", "all");
        var digitalChCount = broadcast.getChannelCount("tv", "DIGITAL");
        var analogChCount = broadcast.getChannelCount("tv", "ANALOG");
        htmlStr1 += "<font style='color:#000000; font-weight:bold'>All Channel
Count:</font> " + channelCount + " [Digital: " + digitalChCount + ", Analog: " +
analogChCount + "];";
        jsLog.lgmethod('getChannelCount()');
        channelList = broadcast.getAllChannelList("tv", "DIGITAL");
        jsLog.lgmethod("getAllChannelList()")

        if (channelList == undefined) {
            htmlStr1 += "<br>No Digital Channel<br>";
        }
        else {
            htmlStr1 += "<br> List of Digital Channel :";
            for (i = 0; i < channelList.length; i++) {
                channelName[i] = channelList[i].channelName;
                signalType[i] = channelList[i].signalType;
                physicalNum[i] = channelList[i].physicalNum;
                if (broadcast.isDvb()) {
                    jsLog.lgmethod('isDvb()');
                    logicalNum[i] = channelList[i].logicalNum; // For DVB
                    htmlStr1 += "<br> [" + (i + 1) + "] Channel Name: " + channelName[i]
+ "<br>Signal Type: " + signalType[i] + "<br>Physical Number: " + physicalNum[i] +
"<br> Logical Number: " + logicalNum[i] + " ";
                }
                else {
                    majorNum[i] = channelList[i].majorNum; // For ATSC
                    minorNum[i] = channelList[i].minorNum;
                    htmlStr1 += "<br> [" + (i + 1) + "] Channel Name: " + channelName[i]
+ "<br> Signal Type: " + signalType[i] + "<br> Physical Number: " + physicalNum[i]
+ "<br>Major Number: " + majorNum[i] + "<br> Minor Number: " + minorNum[i] + " ";
                }
            }
            msgArea.innerHTML = htmlStr1 + "\n";
        }
    }
    showPropResult();
}

function channelListHandler() {

    var channelList;
    var channelCount;
    var channelName = new Array();
    var signalType = new Array();
    var physicalNum = new Array();
    var majorNum = new Array();

```

```

var minorNum = new Array();
var logicalNum = new Array();
var channelCount = broadcast.getChannelCount("tv", "all");
var digitalChCount = broadcast.getChannelCount("tv", "DIGITAL");
var analogChCount = broadcast.getChannelCount("tv", "ANALOG");
htmlStr1 += "<font style='color:#000000; font-weight:bold'>All Channel
Count:</font> " + channelCount + " [Digital: " + digitalChCount + ", Analog: " +
analogChCount + " ]";
jsLog.lgmethod('getChannelCount()');
channelList = broadcast.getChannelList("tv", "DIGITAL", 1);
jsLog.lgmethod("getChannelList()")

if (channelList == undefined) {
    htmlStr1 += "<br>No Digital Channel<br>";
}
else {
    htmlStr1 += "<br> List of Digital Channel :";
    for (i = 0; i < channelList.length; i++) {
        channelName[i] = channelList[i].channelName;
        signalType[i] = channelList[i].signalType;
        physicalNum[i] = channelList[i].physicalNum;
        if (broadcast.isDvb()) {
            jsLog.lgmethod('isDvb()');
            logicalNum[i] = channelList[i].logicalNum; // For DVB
            htmlStr1 += "<br> [" + (i + 1) + "] Channel Name: " + channelName[i]
+ "<br>Signal Type: " + signalType[i] + "<br>Physical Number: " + physicalNum[i] +
"<br> Logical Number: " + logicalNum[i] + " ";
        }
        else {
            majorNum[i] = channelList[i].majorNum; // For ATSC
            minorNum[i] = channelList[i].minorNum;
            htmlStr1 += "<br> [" + (i + 1) + "] Channel Name: " + channelName[i]
+ "<br> Signal Type: " + signalType[i] + "<br> Physical Number: " + physicalNum[i]
+ "<br>Major Number: " + majorNum[i] + "<br> Minor Number: " + minorNum[i] + " ";
        }
        msgArea.innerHTML = htmlStr1 + "\n";
    }
}
showPropResult();

}

function setChannelHandler() {
    var channelList;
    var channelCount;
    var channelName;
    var signalType;
    var physicalNum;
    var majorNum;
    var minorNum;
    var logicalNum;
    var channelToSet = 2;

    channelList = broadcast.getAllChannelList("tv", "DIGITAL");
    jsLog.lgmethod("getAllChannelList()")
    if (channelList == undefined) {
        channelList = broadcast.getAllChannelList("tv", "ANALOG");
        jsLog.lgmethod("getAllChannelList()")
    }

    channelName = channelList[channelToSet].channelName;
    signalType = channelList[channelToSet].signalType;
    physicalNum = channelList[channelToSet].physicalNum;
    if (broadcast.isDvb()) {
        logicalNum = channelList[channelToSet].logicalNum; // For DVB,
        broadcast.setChannel(signalType, physicalNum, logicalNum);
        jsLog.lgmethod("setChannel()")
    }
}

```

```

    }
    else {
        majorNum = channelList[channelToSet].majorNum; // For ATSC,
        minorNum = channelList[channelToSet].minorNum;
        broadcast.setChannel(signalType, physicalNum, majorNum, minorNum);
        jsLog.lgmethod("setChannel()")
    }
}

function programListHandler() {

    var today = new Date();
    var tYear = today.getYear() + 1900;
    var tMonth = modNumberToText(today.getMonth() + 1);
    var tDate = modNumberToText(today.getDate());
    var tHour = modNumberToText(today.getHours());
    var tMin = modNumberToText(today.getMinutes());
    var tSec = modNumberToText(today.getSeconds());
    var start_date = tYear + tMonth + tDate + tHour + tMin + tSec;
    //YYYYMMDDHHmss
    var end_date = tYear + tMonth + tDate + "235959"; //YYYYMMDDHHmss

    var programCount = broadcast.getProgramCount(start_date, end_date);
    jsLog.lgmethod('getProgramCount()');
    htmlStr1 += "<font style='color:#000000; font-weight:bold'>All Program Count:
</font>" + programCount + "<br>";
    var programList = broadcast.getProgramList(start_date, end_date);
    jsLog.lgmethod('getProgramList()');
    var title = new Array();
    var startTime = new Array();
    var endTime = new Array();
    var description = new Array();

    if (programCount != 0) {
        htmlStr1 += "Program List: <br>";

        for (i = 0; i < programList.length; i++) {
            title[i] = programList[i].title;
            startTime[i] = programList[i].startTime;
            endTime[i] = programList[i].endTime;
            description[i] = programList[i].description;
            htmlStr1 += "[" + i + "] Title: " + title[i] + "<br>Start Time: " +
            startTime[i] + "<br>End Time: " + endTime[i] + "<br>Description: " + description[i]
            + " <br>";
            msgArea.innerHTML = htmlStr1 + "\n";
        }
        showPropResult();
    }

}

function currentProgramInfoHandler() {
    var currentProgram = broadcast.getCurrentProgram();
    jsLog.lgmethod('getCurrentProgram()');
    htmlStr1 += "Title: " + currentProgram.title + "<br>Start Time: " +
    currentProgram.startTime + "<br>End Time: " + currentProgram.endTime + "<br>
    Description: " + currentProgram.description + "<br> Event Id: " +
    currentProgram.eventId + " ";
    showPropResult();
}

function nextProgramInfoHandler() {
    var nextProgram = broadcast.getNextProgram();
    jsLog.lgmethod('getNextProgram()');
    htmlStr1 += "Title: " + nextProgram.title + "<br>Start Time: " +
    nextProgram.startTime + "<br>End Time: " + nextProgram.endTime + "<br> description:
    " + nextProgram.description + "<br> Event Id: " + nextProgram.eventId + " ";
    showPropResult();
}

```



```

}

function setSizeHandler() {
    if ((broadcast.width == 240) && (broadcast.height == 180)) {
        broadcast.width = 320;
        broadcast.height = 240;
    }
    else {
        broadcast.width = 240;
        broadcast.height = 180;
    }
    jsLog.lgproperty('width:' + broadcast.width);
    jsLog.lgproperty('height:' + broadcast.height);
}

function checkNetCastVersion() {

    var nBrowserVersion = getBrowserVersion();
    if (nBrowserVersion == 4) // NetCast 2.0
    {
        document.getElementById('enable_numberkey').style.visibility = "hidden";
        document.getElementById('disable_numberkey').style.visibility = "hidden";
        document.getElementById('enable_channelKey').style.visibility = "hidden";
        document.getElementById('disableChannelKey').style.visibility = "hidden";
    }
    else if (nBrowserVersion >= 5) // NetCast 3.0 {
setChannelHandler);
programListHandler);
currentProgramInfoHandler);
nextProgramInfoHandler);
allChannelListHandler);
channelListHandler);
setSizeHandler);

        broadcast.onchannelchange = processChannelChangeFunction;
        broadcast.onnosignal = processNoSignalFunction;
        broadcast.onchannelstatechange = processChStateChangeFunction;
        //broadcast.onepgupdate=processEpgupdate;

        setInnerTextById("method",
"setChannel(),getChannelState()<br>getCurrentChannelName(),getCurrentChannelNumber(
)<br>
getAllChannelList(),getChannelList()<br>getChannelCount(),getCurrentProgram()<br>ge
tNextProgram(),getProgramList()
<br>getProgramCount(),isChannelMapEmpty()
<br>isTunerInput(),isDvb()");
        setInnerTextById("property", "width<br>height");
        setInnerTextById("events",
"onchannelchange<br>onnosignal<br>onchannelstatechange");
    }
}

function modNumberToText(num) {

    if (num < 10) num = "0" + num;
    return num;
}

</script>
</head>
<body ondragstart='return false' onselectstart='return false'>

```

```

<!-- title -->
<div class='SuiteTitle'>LG Smart TV SDK | Web Open API Tutorial</div>
<!-- navigation -->
<div class='SuiteNavigation'>
    <div style="float:left;">File : broadcast/broadcast.html</div>
</div>
<div class='SuiteTitleLine'></div>
<!-- test contents -->
<div id='content_body' class='ContentArea'>
    <div class='ApiListTitleArea'>API List</div>
    <div class='ApiListArea'>
        <div class='MethodTitleArea'>Methods
            <div class='MethodListArea' id="method">channelUp(),
channelDown()
                <br>
            </div>
        </div>
        <div class='PropertyTitleArea'>Properties
            <div class='PropertyListArea' id="property"></div>
        </div>
        <div class='EventTitleArea'>Events
            <div class='EventListArea' id="events"></div>
        </div>
    </div>
    <div class='ViewTitleArea'>
        <div id='tabViewArea' class='SelectedViewArea' style='float:left;'
onclick="showView();">View</div>
        <div id='tabCodeArea' class='UnselectedViewArea'
style='float:right;'
onclick="showCode();">Source</div>
    </div>
    <div id='view'>
        <div class='ViewArea'>
            <table>
                <tr>
                    <td style="width:350px; vertical-align:top">
broadcast" width=240
                        <object id="broadcast" type="application/x-netcast-
                            height=180"></object>
                    </td>
                    <td>
                        <table width="543" border="0">
                            <tr>
                                <td width="163">
class="executeMiddleButton">Channel Up</div>
                                </td>
                                <td rowspan="11" width="380"
style="vertical-align:top;">
                                    <div id="displayArea"
class="displayAreaStyle" style="overflow:auto;height:320px;"></div>
                                    </td>
                            </tr>
                            <tr>
                                <td>
class="executeMiddleButton">Channel Down</div>
                                </td>
                            </tr>
                            <tr>
                                <td>
class="executeMiddleButton">Set Channel</div>
                                </td>
                            </tr>
                            <tr>
                                <td>

```

```

                                <div id="program_list"
class="executeMiddleButton">Program List</div>
                                </td>
                                </tr>
                                <tr>
                                    <td>
                                        <div id="current_programInfo"
class="executeMiddleButton">Current Program</div>
                                        </td>
                                </tr>
                                <tr>
                                    <td>
                                        <div id="next_programInfo"
class="executeMiddleButton">Next Program</div>
                                        </td>
                                </tr>
                                <tr>
                                    <td>
                                        <div id="all_channelList"
class="executeMiddleButton">All Channel List</div>
                                        </td>
                                </tr>
                                <tr>
                                    <td>
                                        <div id="channelList"
class="executeMiddleButton">Channel List</div>
                                        </td>
                                </tr>
                                <tr>
                                    <td>
                                        <div id="set_size" class="executeMiddleButton">Set
Size</div>
                                        </td>
                                </tr>
                                <tr>
                                    <td></td>
                                </tr>
                                <tr height="50px">
                                    <td colspan="2" align="left">
                                        <div id="APIDescription"></div>
                                    </td>
                                </tr>
                                </table>
                                </td>
                                </tr>
                            </table>
                        </div>
                        <div style="visibility: hidden" id='codeview'>
                            <textarea class="SourceCodeArea" value=""
id='sourcecode'></textarea>
                        </div>
                    </div>
                    <!-- button and copyright -->
                    <div class='SuiteButtonArea'>
                        <!-- button -->
                        <div id='btn_back' class='buttonDescription'>BACK</div>
                        <!-- exit key description -->
                        <div id='btn_exit' class='buttonDescription'>EXIT</div>
                        <!--<div id='btn_yellow' class='buttonDescription yellowColor'>NEXT
PAGE</div>-->
                        <!-- copyright -->
                        <div class='copyright'>Copyright LG Electronics</div>
                    </div>
                </body>
            </html>

```