

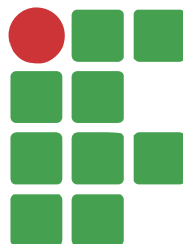
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DEPARTAMENTO DE TELEMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Arquitetura de Suporte à Adaptação Dinâmica de *Apps* de Segunda Tela

Carlos Eduardo Viana Valente

FORTALEZA – CEARÁ

- 2017 -



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
DEPARTAMENTO DE TELEMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Arquitetura de Suporte à Adaptação Dinâmica de *Apps* de Segunda Tela

Autor

Carlos Eduardo Viana Valente

Orientador

Prof. Dr. Cidcley Teixeira de Souza

*Dissertação de Mestrado apresentada
à Coordenação do Curso de Pós-
Graduação em Ciência da
Computação do Instituto Federal de
Educação, Ciência Tecnologia do
Ceará e como parte dos requisitos
para obtenção do grau de **Mestre em
Ciências da Computação**.*

FORTALEZA – CEARÁ

- 2017 -

CARLOS EDUARDO VIANA VALENTE

UMA ARQUITETURA DE SUPORTE À ADAPTAÇÃO DINÂMICA DE APPS
DE SEGUNDA TELA

*Dissertação de Mestrado apresentada
à Coordenação do Curso de Pós-
Graduação em Ciência da
Computação do Instituto Federal de
Educação, Ciência Tecnologia do
Ceará e como parte dos requisitos
para obtenção do grau de **Mestre em
Ciências da Computação**.*

Aprovado em:

BANCA EXAMINADORA

Prof. Dr. Cidcley Teixeira de Souza
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Profa. Dra. Carina Teixeira de Oliveira
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Prof. Dr. Henrique Sérgio Lima Pequeno
Universidade Federal do Ceará (UFC)

Resumo

A dinâmica das redes sociais tem levado cada vez mais os telespectadores de TV interagirem enquanto assistem seus programas prediletos. Essas mesmas redes sociais têm utilizado enquetes, chats, notícias e outros mecanismos para dialogar mais diretamente com esses telespectadores. Contudo, as redes sociais fornecem apenas um meio parcial de interação para esse tipo de mídia, visto que as mesmas não foram criadas para lidar com as peculiaridades dos programas de TV. Para esse fim foram criadas as aplicações de Segunda Tela. Segunda tela pode ser entendida como uma extensão da tela da TV na qual quaisquer informações ou interações podem ser apresentadas aos usuários através de *apps* para *smartphones* ou *tablets*. Diferente das redes sociais, os *apps* de Segunda Tela são disponibilizados pelas emissoras especificamente para usuários de TV, normalmente um *app* para cada programa. Essa alta especificidade desse tipo de *app* gera dois inconvenientes. Para as emissoras é necessário o desenvolvimento de diversos *apps* para cobrir os diversos programas da sua grade, o que gera um custo para a empresa, fato que colabora para explicar o uso massivo de redes sociais por essas emissoras para interagir com os telespectadores atualmente. Na visão dos telespectadores tem-se o inconveniente de ter que instalar diversos *apps* para cada programa, o que acarreta custo de *download* e consumo de recursos dos dispositivos com vários *apps* com finalidades bastantes semelhantes. Argumentamos nessa pesquisa que todos esses *apps* de Segunda Tela poderiam ser incorporados em um único *app* genérico, capaz de realizar dinamicamente adaptações em suas funcionalidades de modo a fornecer um produto customizado para quaisquer emissoras ou programas que queiram explorar os benefícios da Segunda Tela. Para tanto, essa pesquisa apresenta uma arquitetura de suporte a atualização dinâmica de um *app* genérico de Segunda Tela. Essa arquitetura possibilita a implantação de emissoras, programas, mídias, *quizes* e quaisquer outras funcionalidades de Segunda Tela sem a necessidade de criar novos *apps* pelas emissoras nem de instalar novos *apps* pelos usuários. A proposta é validada com estudo de caso que demonstra a utilização da arquitetura para lidar com diversos cenários específicos desse tipo de mídia, tais como programas ao vivo, funcionalidades com tempo de utilidade restritos entre outros. Os resultados demonstram diversos benefícios que a versatilidade da arquitetura proposta oferece para todos os envolvidos na cadeia de produção de conteúdo televisivo.

Palavras-chave: Segunda Tela, adaptação dinâmica, aplicações *mobile*, arquitetura.

Abstract

The dynamics of social networks have increasingly driven TV viewers to interact while watching their favorite shows. Such social networks have been using polls, chats, news and other mechanisms in order to dialogue more directly with their viewers. Nevertheless, social networks provide only a partial way of interaction for this type of media, since they were not designed to deal with the peculiarities of TV programs. To such purpose Second Screen applications were designed. Second Screen can be as an extension of TV screen in which any information or interactions can be presented to users through apps for smartphones or tablets. Unlike social networks, Second Screen apps are made available by broadcasters specifically for TV users, and commonly by one app for each TV program. This aforesaid high specificity results in two disadvantages. Firstly, for TV Stations, it is necessary to develop several apps in order to cover the various programs in their schedule, and as a consequence this generates expenses to such companies, which may explain the massive use of social networks by these broadcasters in order to interact with viewers nowadays. Secondly, from the point of view of television viewers there is the disadvantage of them having to install several apps to each different TV program, which forces additional costs of download as well as resources of devices with many apps very similar in their finalities. It is argued, in this research, that all these Second Screen apps could be incorporated in one single generic app capable of dynamically carry out adaptations in its functions, so that it can offer a customized product to any TV broadcast or TV programs which might want to explore the benefits of Second Screen. For this purpose, it is presented in this research, a supporting architecture to a dynamic update of a generic Second Screen app. The aforementioned architecture enables the roll-out of TV broadcasts, TV programs, networks, quizzes, and any other operability of Second Screen, without the necessity of developing new apps by TV broadcasts and neither of installing new apps by users. This proposal is ensured throughout a case study in which the usage of the supporting architecture to deal with a number of specific scenarios related to this type of media, such as live TV programs, functionality with utility of restricted time, amongst others. The results demonstrated several benefits in which the versatility of the proposed architecture offers to all those involved in the chain of television content production.

Keywords: Second Screen, dynamic adaptation, mobile applications, architecture.

*Dedico este trabalho aos meus pais,
filhos e minha esposa Ana Karine
Portela Vasconcelos, por estar sempre
ao meu lado, com todo seu amor,
carinho, base sólida diante dos
momentos difíceis.*

Agradecimentos

A minha esposa Ana Karine, aos meus pais Alice e Pedro, e aos meus filhos Davi Lucas, João Gabriel e Maria Eduarda, que nos momentos de minha ausência dedicados ao estudo e pesquisa, sempre estiveram ao meu lado e entenderam que o futuro é feito da constante dedicação do hoje.

Ao Prof. Dr. Cidcley Teixeira de Souza pela paciência, dedicação, pelo conhecimento que a mim foi transmitido e por ter me preparado para o meio acadêmico.

Ao Prof. Dr. Wendel Rodrigues por apoiar minha entrada no Programa de Pós-Graduação em Ciência da Computação (PPGCC).

Aos colegas do PPGCC pela convivência e o aprendizado que podemos compartilhar e conquistar.

Agradeço a todos os professores por me proporcionar o conhecimento e, a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

Ao Prof, Dr. Pedro Pedrosa Rebouças Filho Coordenador do PPGCC e à Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico do Ceará (FUNCAP) pelo apoio à este trabalho via Edital 05/2014.

“De tudo, ficaram três coisas: a certeza de estarmos sempre começando, a certeza de que é preciso continuar e a certeza de que podemos ser interrompidos antes de terminar. Façamos da interrupção um caminho novo. Da queda um passo de dança, do medo uma escada, do sonho uma ponte, da procura um encontro! E assim terá valido a pena existir! ”

Fernando Sabino

Sumário

Lista de Figuras	viii
Lista de Tabelas.....	x
Lista de Siglas.....	xi
1 Introdução.....	1
1.1 Objetivo Geral	3
1.2 Objetivos Específicos.....	4
1.3 Estrutura da Dissertação	4
2 Trabalhos Relacionados	5
2.1 Arquiteturas e frameworks de Segunda Tela.....	5
2.2 Softwares com Atualização Dinâmica	9
2.3 Resumo	12
3 Funcionalidades de Segunda Tela encontradas em <i>Apps</i> de TV	13
4 Arquitetura Dinâmica de Segunda Tela.....	20
4.1 Dinamismo para Segunda Tela	20
4.2 Arquitetura Proposta.....	21
4.3 Atualizador de Serviços.....	26
4.4 Habilitando ADST para Receber Funções e Conteúdos.....	27
4.5 Padrão para Transferência e Comunicação	30
5 Protótipo.....	34
5.1 Aplicativo Dinâmico de Segunda Tela (ADST)	34
5.2 Administração de Funções para Segunda Tela (AFFST).....	37
6 Estudo de Caso	43
6.1 Cenário para Validação da Arquitetura	43
6.1.1 Validar o Tempo da Entrega do Pacote de Dados da Função Enquete ..	45
7 Conclusão.....	49
7.1 Contribuição	49

7.2	Trabalhos Futuros	50
7.3	Publicação.....	50
	Referências	51
	Anexo A - <i>Checklist</i> de Funcionalidades em <i>Apps</i> de Segunda Tela	58
	Anexo B - <i>Apps</i> de TVs Brasileira e Internacional	60

Lista de Figuras

Figura 2.1 - Arquitetura Básica do projeto Smarties: Setas arredondadas indicam que os celulares clientes se comunicam através da biblioteca, por exemplo, a informação é enviada para a biblioteca que o transforma em um evento para o aplicativo; e envia de volta para os clientes	10
Figura 3.1 - Análise geral dos aplicativos das emissoras que apresentam funções de Segunda Tela	19
Figura 4.1 - Arquitetura de atualização dinâmica ao ADST	23
Figura 4.2 - Dados gravados no Banco de Dados a partir do ADST	25
Figura 4.3 - Modelo do Pacote de Dados para habilitar ADST	27
Figura 4.4 - Etapas para habilitar funções no ADST	28
Figura 4.5 - Pseudocódigo para habilitar ADST.....	29
Figura 4.6 - Modelo de string JSON com conteúdo de Segunda Tela.....	30
Figura 4.7 - Tela da Enquete do ADST renderizada dinamicamente	30
Figura 4.8 - Etapas para Invocar <i>Web Service</i> baseado em JSON.....	32
Figura 5.1 - Tela inicial para selecionar emissora no ADST	35
Figura 5.2 - Tela com informações do programa e funções de Segunda Tela no ADST	35
Figura 5.3 - Tela do menu do o ADST.....	36
Figura 5.4 - Tela de cadastro das emissoras no AFFST	37
Figura 5.5 - Tela de cadastro dos programas das emissoras no AFFST	38
Figura 5.6 - Tela de cadastro da grade de programação das emissoras no AFFST	39

Figura 5.7 - Tela do menu AFFST	39
Figura 5.8 - Tela de cadastro das funções, exemplo da criação da função Enquete para programa da emissora no AFFST.....	40
Figura 6.1 - Etapas para atualização da função Enquete no ADST	44
Figura 6.2 - Análise do recebimento e envio do Pacote de Dados no Atualizador de Serviços para criação da Enquete no ADST	46
Figura 6.3 - Análise do recebimento e envio do Pacote de Dados no Atualizador de Serviços para criação da Enquete no ADST	47

Lista de Tabelas

Tabela 3.1 - Amostras dos aplicativos levantados das emissoras de televisão e subconjuntos do <i>checklist</i>	17
Tabela 3.2 - Somatório das respostas levantadas no <i>checklist</i>	17
Tabela 3.3 - Média das respostas levantadas no <i>checklist</i>	18
Tabela 4.1 - Etapas gerais da arquitetura dinâmica	23
Tabela 4.2 - Etapas do processo das operações de CRUD iniciadas pelo ADST ao Atualizador de Serviços	25
Tabela 4.3 - Descrição das etapas para habilitar funções do ADST	29
Tabela 4.4 - Etapas para o procedimento de invocação de requisição e resposta ao <i>web service</i>	32
Tabela 4.5 - Elementos usados na construção do REST. Fonte Aece (2013) .	32
Tabela 6.1 - Elementos usados na construção do REST	45

Lista de Siglas

API	<i>Application Programming Interface</i>
APP	<i>Application</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
JSON	<i>JavaScript Object Notation</i>
REST	<i>Representational State Transfer</i>
URI	<i>Uniform Resource Identifier</i>
W3C	<i>World Wide Web Consortium</i>
XML	<i>Extensible Markup Language</i>

Capítulo 1

1 Introdução

A experiência de ver televisão é uma atividade envolvente capaz de despertar emoções, aumentar consumismo, favorecer aprendizado, dentre outros efeitos. O que faz com que milhares de telespectadores fiquem frente da TV para assistir jogos, jornais, filmes, novelas, seriados, *realitys*, programas de auditório dentre outros. O comportamento do telespectador está ligado aos processos do entretenimento. Para Murray (2003) deve-se assistir à programação televisiva no aparelho de televisão, mas com oportunidades de se poder comentar, interagir, participar, ter conteúdo adicional.

Com a diversidade de dispositivos móveis como *smartphone* e *tablet* conectados à internet, surgem possibilidades para as emissoras de televisão oferecerem informações de forma interativa e complementar aos telespectadores, criando possibilidades em diversos segmentos de uma forma ágil à uma grande diversidade de público.

Em pesquisa realizada para entender hábitos de uso de TV pela empresa Ericson ConsumerLab¹, onde foram entrevistadas 100.000 pessoas em 40 países no ano de 2012, revelou que 62% das pessoas usam dispositivos móveis enquanto assistem TV e que

¹ <https://www.ericsson.com/networked-society/consumerlab>

40% delas estão comentando o que assistem no momento do programa. Essa pesquisa foi também realizada em 2015 e mostrou que aumentou para 71% o número de pessoas que usam *smartphones* e *tablets* enquanto assistem televisão.

Outra pesquisa realizada pela Nielsen² que estuda o comportamento e hábitos de consumidores ao redor do mundo, em 2015 mostrou que 49% dos usuários no mundo assistem programação televisiva com interação por meio de dispositivos móveis, e que no Brasil esse número chega a 50%.

Se antes o telespectador assistia ao conteúdo televisivo para depois interagir, hoje ele pode fazer as duas ações, sem interferir na experiência coletiva da TV. Com a Segunda Tela os telespectadores de TV têm a possibilidade de interagir com o que é transmitido na Primeira Tela (ZIEGLER, 2013). Os usuários de TV podem acompanhar as programações das emissoras em Segunda Tela, através de aplicativos (*app*) que complementem a grade de programação da TV.

Segunda Tela é o nome dado ao hábito de telespectadores utilizarem *tablets*, *smartphones* ou computador conectados à internet Geerts, Leenheer, *et al.* (2014), através de aplicações específicas que permitem interações com os conteúdos disponibilizados por emissoras de TV (ANGELUCI, 2013).

A interatividade oferecida pela Segunda Tela, complementa a programação que televisionada e amplia a experiência televisiva, por exemplo através de *games*, enquetes, *quiz*, chat, votação ao vivo, compartilhamento em redes sociais, notícias e outros (RUBIO, ELENA e DAVIC, 2014).

“O analista José Calazans, da Nielsen IBOPE, afirma que esse cenário cria oportunidade, enquanto várias telas dão aos espectadores mais opções, eles

² <http://www.nielsen.com/br/pt/press-room/2015/69-dos-consumidores-globais-acreditam-que-interacoes-cara-a-cara-estao-sendo-substituidas-por-eletronicos1.html>

também dão aos provedores de conteúdo e anunciantes mais oportunidades e maneiras de alcançar e engajar os telespectadores.³

Muitas emissoras de TV não possuem aplicativos para dispositivos móveis por questões técnicas ou por conta do custo para se construir e manter a tecnologia. Por outro lado, as emissoras de TV que possuem aplicativos de Segunda Tela não apresentam engajamento com as funcionalidades de Segunda Tela.

Na programação televisiva percebe-se que para cada novo programa é criado um aplicativo, com isso os telespectadores sempre precisam instalar novos *apps*, ocasionando aumento do consumo de memória e de processamento. E, para que os aplicativos sejam atualizados, precisam ser reinstalados ou atualizados pelas lojas *online* de aplicativos.

Com a expansão da tecnologia de Segunda Tela, nessa pesquisa é proposto uma arquitetura de atualização dinâmica que ofereça as emissoras a possibilidade de criar funcionalidades com conteúdo atualizado à programação televisiva. E promova atualização dinâmica dessas funcionalidades com temporização na utilização junto a seus programas gravados e ao vivo, através da entrega dos pacotes de dados com atualizações de Segunda Tela por meio de um Atualizador de Serviços.

1.1 Objetivo Geral

Propor uma arquitetura de gerenciamento de um *app* genérico de Segunda Tela que pode ser adaptado dinamicamente para atender a diversas emissoras de TV e seus programas.

³ Extraído de: <http://www.nielsen.com/br/pt/press-room/2015/69-dos-consumidores-globais-acreditam-que-interacoes-cara-a-cara-estao-sendo-substituidas-por-eletronicos1.html>

1.2 Objetivos Específicos

Criar uma arquitetura de atualização dinâmica de Segunda Tela para um aplicativo *mobile* denominado Aplicativo Dinâmico de Segunda Tela (ADST).

Utilizar um conjunto de tecnologias *web* que permite a aplicação da arquitetura de Segunda Tela, com temporização para utilização dos programas gravados e ao vivo das emissoras de TV.

1.3 Estrutura da Dissertação

Esta dissertação está organizada em cinco capítulos. O Capítulo 1 apresenta uma introdução a Segunda Tela, bem como objetivos da pesquisa. O Capítulo 2 apresenta uma revisão bibliográfica sobre o objetivo da pesquisa. O Capítulo 3 apresenta um levantamento das funcionalidades encontradas em *apps* de TV. No Capítulo 4, descreve-se a arquitetura dinâmica de aplicativos de Segunda Tela e a fundamentação sobre a proposta da arquitetura desenvolvida. O Capítulo 5 apresenta os protótipos para o Aplicativo Dinâmico de Segunda Tela e o Administrador de Funções para Segunda Tela. No Capítulo 6, descreve-se o um estudo de caso para a validação da arquitetura da Segunda Tela. O último capítulo apresenta as conclusões, sugestões para trabalhos futuros e publicação.

Capítulo 2

2 Trabalhos Relacionados

A literatura especializada retrata vários trabalhos com modelos adotados para arquitetura e atualização de funcionalidades em aplicativos de Segunda Tela. No entanto, diferentemente desta pesquisa, muitos utilizam tecnologias de alto custo e possuem pouca dinamicidade na atualização de funcionalidades ou informações enviadas aos aplicativos.

2.1 Arquiteturas e frameworks de Segunda Tela

Em Šoškić, Kuzmanović, *et al.* (2014) foi apresentado um *framework* baseado na arquitetura cliente-servidor para distribuir e apresentar conteúdo extra para TV Digital. Do lado do cliente, estão os dispositivos com o *app* de Segunda Tela nos *smartphones* e *tablets*. Do lado servidor, é implementado um servidor de mídia em uma *set-top box* com Sistema Operacional *Android*. A interação básica entre os dois lados da arquitetura consiste em um servidor para receber e processar o conteúdo enviado pelos difusores e, em seguida, distribuir o conteúdo para os dispositivos dos clientes. O servidor protege o conteúdo extra recebido, logo ele é capaz de separar o conteúdo em duas categorias: livre e *premium*.

Em Soskic, Tapavicki e Rasic (2014) foi relatado a utilização de um controle remoto para dispositivos com Sistema Operacional *Android*. O aplicativo permite que o usuário assista a programas da TV pelo dispositivo móvel, independentemente da televisão. Sem dinamismo para atualização automática de funcionalidades de Segunda Tela, o aplicativo apenas exibe o que é transmitido na TV para o dispositivo móvel.

A proposta apresentada por Punt, Bjelica, *et al.* (2015) foi um *framework* para concentrar jogos a serem executados durante a transmissão de um programa de TV. Os *smartphones* são usados como controle remoto e para a visualização de mais informações sobre o jogo. O artigo relata o desenvolvimento de micro jogos para utilização durante a transmissão de um programa esportivo. Entretanto, os jogos ocupam espaço na Primeira Tela, tirando a atenção da programação que está sendo exibida e focando no jogo, Além de não apresentar dinamismo para a criação automática de novas funcionalidades. O *app* proposto funciona somente como um controle remoto para os jogos.

Em Ziegler (2013) foi apresentado o desenvolvimento de um *framework* para prover comunicação bidirecional entre aplicações *web* para TV e dispositivos de Segunda Tela. O *framework* encontra, conecta e realiza a comunicação entre as duas telas. Nesse modelo, apenas uma emissora é acessada por um *browser* ou aplicativo. A Segunda Tela é apenas uma cópia da Primeira, com funções estáticas.

Em Sutter, Matton, *et al.* (2011) tratou-se do conceito de *Customer Relationship Management* (CRM) no universo dos conteúdos de rádio e TV. A proposta usa a Segunda Tela para coletar dados dos consumidores a fim de adquirir informações para os difusores e publicitários. Para validar os conceitos, foi criado um sistema de arquitetura escalável para processar as informações do usuário. E para garantir que a Segunda Tela seja detectada apenas quando o conteúdo estiver sendo consumido, é

usada marca d'água no áudio. Porém, a coleta dos dados fica restrita por conta do uso da tecnologia.

É proposto por Arnold, Chen e Baum (2014) um algoritmo cujo código fonte para sincronismo de Segunda Tela é usada marca d'água no áudio. Esse sistema conta com três componentes: modelo psico-acústico, quadro de ressincronização e detector de correlação. No entanto, a eficiência da tecnologia da marca d'água de áudio depende do *hardware* dos transmissores e receptores, sendo, em sua grande maioria, usada para medição de audiência.

Foi proposto por Calixto, Angeluci e Kurashima (2014) uma comparação da eficácia em cenários interativos de mídia específicos, seguida de uma avaliação de transmissão e recepção de áudio no contexto de *Internet Protocol Television* (IPTV). Foi utilizado marca d'água de áudio para sincronismo de conteúdo entre TV e dispositivos de Segunda Tela. Considerou-se que a marca d'água de áudio é uma técnica usada principalmente para autenticação e proteção na transmissão de conteúdo para TV. Os dados coletados neste experimento demonstram que a marca d'água de áudio é adequada para o sincronismo, porém, depende do *hardware* dos transmissores e receptores.

Em Howson, Gautier, *et al.* (2011) foi proposto um sistema baseado no trabalho *Synchronized and scalable AV Across Networks* (SAVANT) para sincronizar sobre *broadcast* e componentes audiovisuais fornecidos por diferentes protocolos de transporte. Nele, o telespectador assiste a um programa acompanhado de áudio adicional através do dispositivo móvel, podendo ouvir outros idiomas do que é transmitido pela TV. No entanto, para a sua utilização se faz necessário o uso de *hardware* específico.

Uma solução para o uso de Segunda Tela apresentada por Rodrigues, Reis e Lucena (2014), foi o protocolo chamado de *Transmission Transport Stream*. Ele visa

permitir aos usuários o acesso a interação com conteúdo adicional através da internet. Vale ressaltar que o conteúdo de Segunda Tela pode estar sincronizado com a transmissão ao vivo da TV. O propósito do trabalho foi facilitar o acesso e a interação de dispositivos móveis a conteúdos de Segunda Tela providos pelos difusores.

Em Seruffo, Miranda, *et al.* (2015) foi desenvolvido o aplicativo Dengue fever, que utiliza uma sincronização de dispositivos móveis como tablet e smartphones com aparelhos de TV digital, a partir da leitura de QR code. O app possui jogos interativos, tendo como único foco o combate do mosquito *Aedes aegypti*. O app não apresenta um framework que possibilite a criação de novas funções dinamicamente para os usuários.

Em Cortez, Shamma e Cai (2012) apresentam um protocolo para viabilizar novas experiências em interação da TV Digital com dispositivos de Segunda Tela. Uma das motivações da modelagem do novo protocolo são as normas da W3C para interação com Segunda Tela, que na avaliação dos autores, está desconectada com as necessidades dos sistemas comerciais. O serviço proposto pelo protocolo se resume a interação do dispositivo móvel com a TV, usando uma nuvem de dados para permitir a busca e reconhecimento de dispositivos na rede.

Em Segundo e Santos (2014) utilizou-se a revisão sistemática, para abordar o sincronismo entre múltiplos conteúdos, em uma TV. A pesquisa combina métodos para mapeamento e revisão sistemática de literatura para investigar o estado corrente da pesquisa no assunto supracitado. A importância desse trabalho é permitir a comunidade de pesquisadores em TV Digital conhecer o estado da arte em sincronismo de múltiplos conteúdos em TV até 2013.

Diante dos trabalhos retratados até aqui, percebe-se que os mesmos utilizaram tecnologias, técnicas e equipamentos de alto custo, além de não possuírem métodos dinâmicos de atualização de aplicativos.

2.2 Softwares com Atualização Dinâmica

Verificou-se que, na literatura existem trabalhos que não abordam aplicativos de Segunda Tela. Tratam apenas de atualizações dinâmicas em *softwares* de um modo geral, como em Louberry, Roose e Dalmau (2011), que traz um *framework* chamado Kalimucho, o qual se propõe atualizar dinamicamente *softwares* que estão na mesma rede de computadores, diferentemente do que se propõe nesta pesquisa.

Em Hicks e Nettles (2005) foi proposto um *framework* para atualização dinâmica de sistemas através de um servidor *web* denominado *FlashEd*, que atualizava códigos em linguagem C. Esse projeto foi limitado por ter uso exclusivo em rede local, e para aplicações específicas da linguagem, não permitindo aplicações diferentes.

Em Chen, Jin, *et al.* (2016) foi relatado *softwares* de atualização dinâmica e proposto o sistema *Replus*. Para atualização dinâmica de *softwares*, apresenta técnicas que utilizam pacotes de comandos em estrutura de dados para Sistemas Operacionais. Eles corrigem programas em tempo de execução sem serem reiniciados. Fica restrito a *softwares* específicos, ele não implementa novas funções, e sim corrige funcionalidades já existentes.

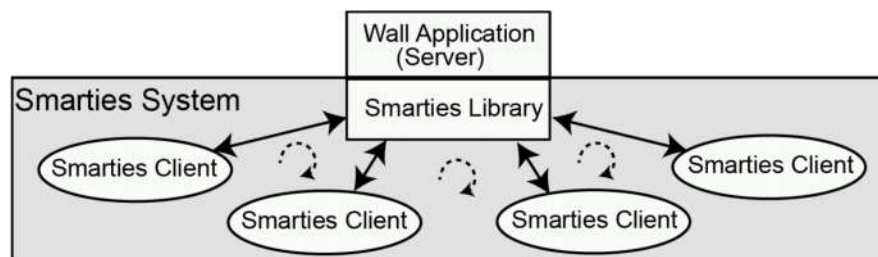
É apresentado em Ashmore e Makki (2011) o funcionamento, *designer* e especificações de um *middleware* chamado de IMISSAR, com a finalidade de aumentar a performance da execução dos aplicativos nos dispositivos móveis. Captura erros ocorridos nas transmissões dos dados e serviços até o dispositivo móvel. Possui um recurso chamado de *reconfiguração automática* que controla a entrega dos dados aos aplicativos dos dispositivos móveis, que permite o dispositivo mudar a aplicação para ser executada remotamente ao invés de executar localmente.

O *middleware* chamado de IMISSAR transfere processos de um aplicativo em execução de um dispositivo móvel para um servidor remoto citado em Murarasu e Magedanz (2009). Avalia fatores como: a vida da bateria, o consumo da bateria atual,

o uso do processador, uso de memória, as preferências do usuário e conexão de rede, para que decida se a aplicação será executada localmente ou remotamente.

É apresentado em Chapuis, Bezerianos e Frantzeskakis (2014) o projeto *Smarties* com uma estrutura de código aberto, protocolo de comunicação entre os dispositivos e a aplicação da Primeira Tela. Foca no dispositivo móvel, com sistema integrado a um alto nível de protocolo, juntamente com bibliotecas que escondem a complexidade do protocolo. O protocolo de comunicação possui arquitetura de software que consiste em uma aplicação executada em uma máquina (servidor) ligada aos dispositivos móveis (clientes). A entrada do usuário é enviada para a máquina (servidor), que instrui os dispositivos móveis (clientes) que modificam seu estado de renderização. Com isso, são recebidas nos dispositivos móveis (clientes) com mostra a Figura 2.1.

Figura 2.1 - Arquitetura Básica do projeto Smarties: Setas arredondadas indicam que os celulares clientes se comunicam através da biblioteca, por exemplo, a informação é enviada para a biblioteca que o transforma em um evento para o aplicativo; e envia de volta para os clientes



Fonte: Chapuis, Bezerianos e Frantzeskakis (2014)

Em Krone, Müller e Ertl (2015) é apresentado um aplicativo para dispositivos com sistema operacional *Android* equipados com unidades de processamento gráfico (GPUs) com aceleração por hardware OpenGL ES renderização. Capaz de realizar visualização e manipular remotamente imagens científicas que são apresentadas ao mesmo tempo em outra tela. O *app* estabelece uma conexão com o servidor por meio da leitura de um QR code que é gerado dinamicamente pelo servidor.

Em Ilvonen, Ihantola e Mikkonen (2016) realizou-se uma revisão de literatura a partir dos guias de engenharia de *software*, como o *Software Engineering Body of*

Knowledge (SWEBOK) e o *Software Engineering Education Knowledge* (SEEK), a respeito de técnicas para sistemas de atualização dinâmica de *software*. Concluiu-se que são necessários mais estudos para possíveis experiências de utilização da atualização dinâmica em engenharia de *software*.

Em Zheng, Ormandjieva e Fan (2015) o trabalho tem foco em adaptações dinâmicas em âmbito de design e usabilidade em um aplicativo *e-commerce* denominado ZOE. Esse, aborda a manipulação das informações relacionadas à interface do usuário móvel, como, por exemplo, tamanhos de fontes, cor da tela, brilho, nitidez e volume. As adaptações dinâmicas do aplicativo não ocorrem de forma *on-line*. O *app* não recebe pacotes de dados que façam que ele seja alterado dinamicamente.

Em Bisignano, Di Modica e Tomarchio (2005) foi apresentado uma implementação que permite aos desenvolvedores especificar interações do usuário com o aplicativo. Realiza adaptações dinâmicas através do ajuste da interface do usuário para recursos de imagens, vídeos e áudios com o uso da linguagem XML. A aplicação realiza renderizações compatíveis com a interface do usuário, facilitando a usabilidade e evitando consumos de recursos de rede.

Em Chmielewski, Flotyński, *et al.* (2016) propõe-se uma linguagem de descrição da interface do usuário (UIDL) que retrata vários aspectos de uma interface de usuário (IU) para facilitar a análise e o desenvolvimento de aplicativos que podem ser disponibilizados aos usuários através de qualquer dispositivo. Traz um conjunto de componentes pré-definidos para serem renderizados em plataformas diferentes.

Na pesquisa de Castillejo, Almeida e López-De-Ipiña (2014) foi realizada a revisão do estado da arte em sistemas de interface de usuário adaptáveis nos últimos 20 anos. Com um modelo que combina três entidades principais (usuários, contexto e dispositivos) representados em ambientes adaptáveis. Ademais, é proposto um modelo para projetar modelos de interface de usuário adaptáveis, o qual inclui: as

características dos usuários divididas em categorias (interface, áudio, exibição e visão), contexto (primário: física, alto nível; e secundário: metadados do ambiente, ambiente virtual) e dispositivo (disponibilidade atual, nível de bateria, conectividade de rede, pontos de acesso de *wi-fi*, status do dispositivo, brilho da tela, nível do som, aceleração e orientação).

2.3 Resumo

Diante do cenário descrito, o objetivo desta pesquisa é criar uma arquitetura de suporte capaz de atualizar dinamicamente aplicativos de Segunda Tela através de um conjunto de tecnologias *web* usadas para o funcionamento de um Aplicativo Dinâmico de Segunda Tela (ADST). Propicia, através da *internet*, que os dispositivos móveis recebam pacotes de dados de forma dinâmica e atualizada. Demonstrando a utilização da arquitetura para lidar com diversos cenários específicos, tais como programas ao vivo, funcionalidades com tempo de utilidade restrito, dentre outros. Dessa forma, não existindo necessidade de enviar novas versões pelas lojas de aplicativos quando houver atualizações.

Capítulo 3

3 Funcionalidades de Segunda Tela encontradas em *Apps* de TV

Conforme é possível verificar no “Anexo A” deste trabalho a partir de Nahuz e Santos (2015), foi formulado um *checklist* com 20 (vinte) perguntas para avaliar aplicativos de emissoras de televisão brasileira e internacional apresentadas no “Anexo B” quanto a uso da tecnologia de Segunda Tela. Para o *checklist* foram utilizadas técnicas de engajamento de Askwith (2007) que trata: a relação público e televisão, plataformas de mídia, conteúdos, produtos, redes sociais e oportunidades de envolvimento do público com o conteúdo, com foco na expansibilidade do conteúdo televisivo.

Os engajamentos são divididos em categorias das quais foram selecionadas 07 (sete) para categorizar os aplicativos de Segunda Tela, também citadas em Santos e Nahuz (2015), são elas:

- Acesso Expandido - refere-se às oportunidades de envolvimento que resultam da distribuição do conteúdo original através de múltiplas plataformas, dispositivos e modelos de negócios;

- Conteúdo Reembalado - refere-se às oportunidades para o envolvimento com o conteúdo original reeditado de outra forma. Não fornece novas informações sobre a narrativa. Seu propósito é guiar o espectador casual e reavivar o interesse do fã, garantido a ele o prazer do reconhecimento (são exemplos os guias de episódios ou de temporadas, biografias dos personagens e recapitulações);
- Conteúdo Expandido - oportunidades de engajamento através de conteúdo adicional desenvolvido para aumentar, contextualizar, complementar ou prover informações sobre o texto televisivo, incluindo extensões textuais (divididas em extensões narrativas, que oferecem novas histórias nos formatos típicos da narrativa comercial e extensões diegéticas, conteúdos que saem do universo narrativo), informação relevante (que nutre o universo com detalhes que compõem uma espécie de conhecimento enciclopédico dentro do seu contexto) e informação extratextual (sendo dividida em informação industrial, dados profissionais sobre a equipe e sobre a produção e em informação de celebridades ligadas ao projeto);
- Produtos de Marca - oportunidades de engajamento que enfatizam práticas de aquisição e propriedade pelo telespectador, proporcionadas pela produção de coisas relacionadas ao programa, itens que podem ser gratuitos ou vendidos. Esses produtos servem como recordações do programa e indicam publicamente a afiliação do proprietário àquele universo narrativo;
- Atividades Relacionadas - oportunidades de engajamento que posicionam o espectador como um participante ativo em atos relacionados com o programa, incluindo atividades temáticas, de experiência produtiva e atividades de desafio;

- Interação Social - engajamento que posiciona o texto televisivo como base ou pretexto para a interação com outros telespectadores (interação horizontal), com profissionais criativos do programa (interação vertical), ou com personagens da narrativa (interação diagonal);
- Interatividade - oportunidades de engajamento que permitem alguma interação direta entre o telespectador-participante e o programa de televisão. Aqui o autor divide essa interação, ou seja, essa troca de ações e reações, em quatro possibilidades: a interação mecânica (as interações físicas e materiais com o dispositivo, como trocar de canal), a ativação de conteúdo (o usuário escolhe entre uma série de ações pré-determinadas para acionar uma reação pré-programada, como participar de enquetes), a interação de conteúdo (sendo dividida em contribuições reconhecidas, que apresentam uma estrutura específica para que o espectador possa fazer uma contribuição e ter essa contribuição reconhecida, e em interações influentes, onde as interações do público interferem no resultado do programa) e a interação social (explicada no item anterior).

Ainda em Askwith (2007) é dividida as relações estabelecidas do telespectador com o conteúdo televisivo em cinco Lógicas de Engajamento, as quais também são descritas em Nahuz e Santos (2015), são elas:

- Lógica do Entretenimento, a necessidade do telespectador de ser entretido pelo programa de televisão em outros níveis além da experiência tradicional de assistir à TV;
- Lógica da Conexão Social, a necessidade do telespectador de se utilizar do programa de televisão para se conectar com amigos, familiares e outros telespectadores;

- Lógica do Domínio, a necessidade do telespectador de exercer domínio sobre a narrativa do programa de televisão, interpretando informações e conteúdos adicionais de forma a compreender as relações textuais e os formatos narrativos utilizados no programa para gerar suas próprias opiniões e teorias sobre a trama do programa;
- Lógica da Imersão, a necessidade do mesmo de enxergar o universo ficcional como um mundo lógico e possível, utilizando todos os conteúdos adicionais para preencher a sensação de pertencimento à trama, ao programa de televisão, e/ou à equipe que o produz;
- Lógica da Identificação, que se apresenta como a necessidade do telespectador se identificar, para si ou para outros, a partir do programa de televisão, seja apenas analisando e comparando sua própria personalidade, seus valores, e seus ideais ao que é proposto pelo programa, ou também assumindo uma associação dessas características ao próprio programa em um contexto social, uma espécie de afiliação ao programa televisivo.

De posse das respostas do *checklist* foi realizada uma análise visando a identificação de aspectos que ainda não foram abordados na literatura. Para identificar o perfil para aplicativos de Segunda Tela com interface e funcionalidades que poderão ser customizadas e atualizadas a partir de uma arquitetura que será proposta aqui nesta pesquisa.

Os resultados que serão apresentados foram possíveis a partir da amostragem do levantamento dos aplicativos de Segunda Tela das emissoras de televisão. Assim serão apresentados dados quantitativos levantados a partir do *checklist* apresentado no “Anexo A”. Com isso, podem ser qualificados e medidos a relevância que os aplicativos apresentam em relação a Segunda Tela e, traçar um padrão para aplicativo de Segunda Tela que será proposto nessa pesquisa.

Para os resultados das respostas do *checklist* foram usadas as variáveis para dados dicotômicos descritos em Bragança (1977) e Lira e Neto (2006) onde as respostas podem assumir condições de “Sim” ou “Não” que serão mostradas a seguir.

A Tabela 3.1 apresenta as amostras dos aplicativos levantados das emissoras de televisão, e os subconjuntos dos itens respondidos no *checklist*. Apresentando Sim (S) ou Não (N) nas respostas dos itens do subconjunto do *checklist*. Para a validar a análise dos dados foi aplicado o método estatístico por amostragem na média dos dados obtidos no subconjunto dos aplicativos pesquisados.

Tabela 3.1 - Amostras dos aplicativos levantados das emissoras de televisão e subconjuntos do *checklist*

		SUBCONJUNTO																			
AMOSTRAS		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	Globo Play	S	S	S	S	N	N	N	N	N	N	N	N	S	S	N	N	N	S	S	N
	Globo	N	N	N	S	S	N	N	N	N	N	S	N	S	S	N	N	S	N	S	N
	GloboEsporte	S	S	S	N	N	S	S	N	N	N	S	N	N	S	N	N	S	S	S	N
	TheVoiceKids	S	S	S	S	N	N	S	S	N	N	S	S	N	S	N	N	S	N	N	S
	BBB 16	S	N	S	N	N	N	N	N	N	N	S	S	N	S	N	N	S	N	N	S
	SBT	S	S	S	S	S	N	S	S	N	N	N	N	S	N	N	N	N	S	N	N
	Rede Record	N	S	S	N	N	N	S	N	N	N	N	N	S	S	N	N	S	S	S	N
	BAND	S	S	S	S	S	N	S	N	N	N	N	S	S	N	N	N	S	S	N	N
	CNN	S	S	S	N	N	S	N	N	N	N	N	S	S	N	N	N	S	S	N	N
	FOX Play	S	S	S	N	N	N	S	N	N	N	N	N	S	N	N	N	N	S	N	N
	Televisa	S	N	S	N	S	N	S	N	N	N	N	N	S	S	N	N	S	S	N	N
	ESPN Sync	S	N	N	N	N	N	N	N	N	N	S	S	N	N	N	N	S	S	N	N

Fonte: Autor (2017)

Como resultado das respostas levantadas no *checklist*, se obteve o somatório da quantidade de respostas Sim (S) como também as respostas Não (N) de cada item, conforme mostrados na Tabela 3.2.

Tabela 3.2 - Somatório das respostas levantadas no *checklist*

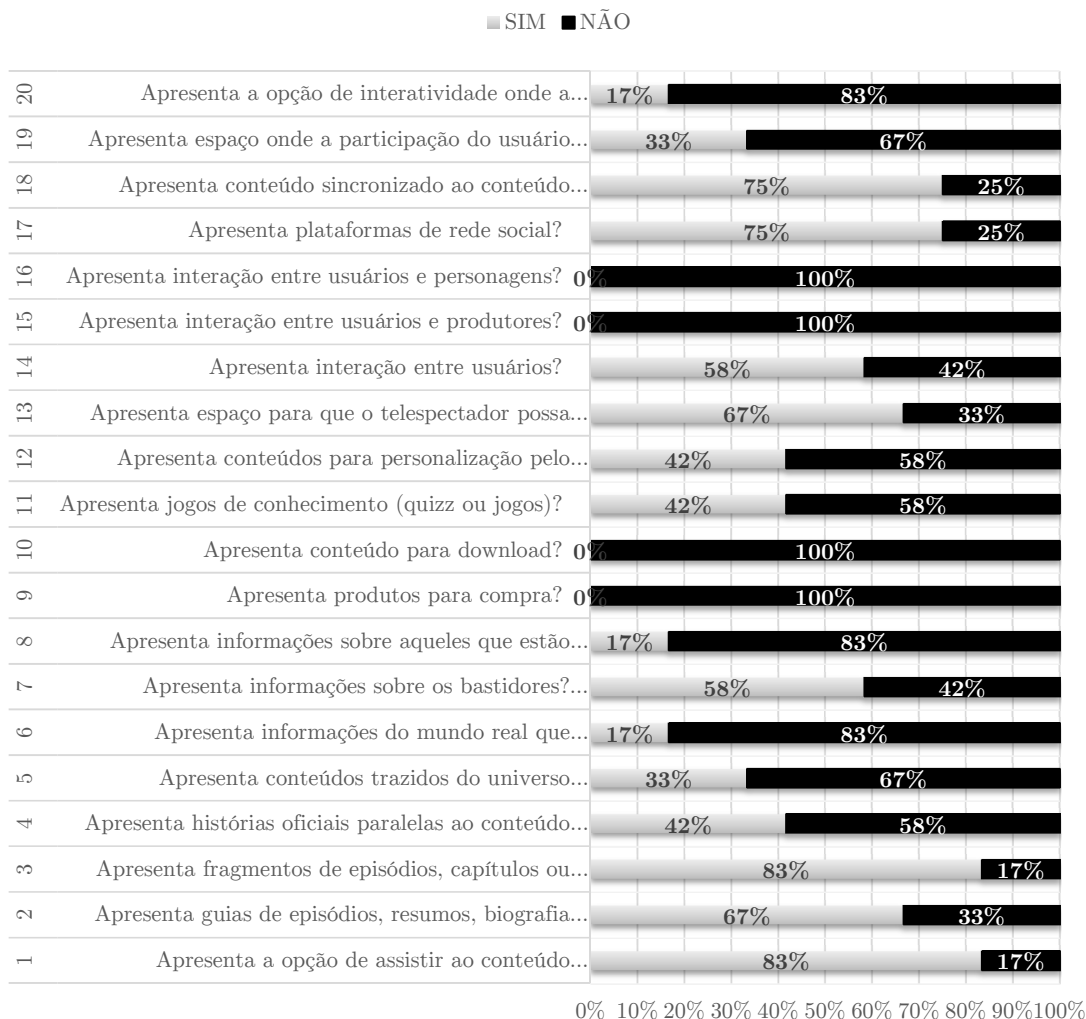
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Total de Sim	10	8	10	5	4	2	7	2	0	0	5	5	8	7	0	0	9	9	4	2
Total de Não	2	4	2	7	8	10	5	10	12	12	7	7	4	5	12	12	3	3	8	10

Fonte: Autor (2017)

No Tabela 3.2 foi observado que a média das respostas aparecem com uma maior relevância em Não (N), o que demonstra que muitos itens do *checklist* não foram atendidos pelos aplicativos das emissoras naquilo que se propõe a Segunda Tela a partir

do *checklist*. Contudo, nessa amostragem mesmo com número menor existem aplicativos que atendem Sim (S) alguns dos itens do subconjunto da Tabela 3.1, conforme percentual mostrado também na Tabela 3.3.

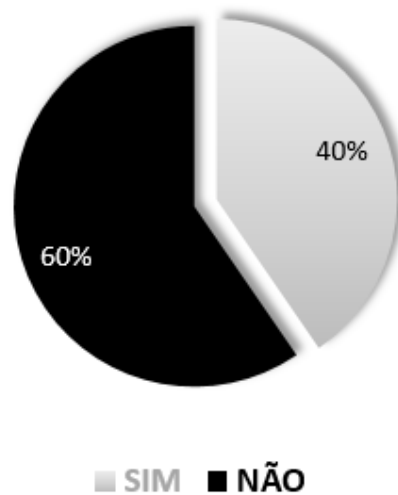
Tabela 3.3 - Média das respostas levantadas no *checklist*



Fonte: Autor (2017)

A diferença entre Sim (S) e Não (N) é bem relevante na média geral das respostas conforme apresentado na Figura 3.1, ao mostrar que 60% das funções dos aplicativos não atendem aos requisitos de Segunda Tela apresentados no *checklist*, onde somente 40% das funções são encontradas nos aplicativos das emissoras.

Figura 3.1 - Análise geral dos aplicativos das emissoras que apresentam funções de Segunda Tela



Fonte: Autor (2017)

Com o resultado proposto a partir das características levantadas no *checklist* para um aplicativo de Segunda Tela, o aplicativo deverá ser coberto por uma arquitetura que permita em um único *app* várias poderem adicionar programas da sua programação televisiva. Onde o telespectador não precise instalar vários aplicativos no seu dispositivo. Com o *app* a emissora poderá adicionar e remover suas programações e suas respectivas funções de Segunda Tela em sincronismo para atualização com o que é transmitido na Primeira Tela, com regras específicas para quando e o tempo que as funcionalidades estarão disponíveis no aplicativo.

Capítulo 4

4 Arquitetura Dinâmica de Segunda Tela

4.1 Dinamismo para Segunda Tela

Chama-se sistema de Atualização Dinâmica de *Software* ou *Dynamic Software Updating* (DSU), o sistema que permite que um programa seja alterado instantaneamente enquanto ainda está em execução (ILVONEN, IHANTOLA e MIKKONEN, 2016). Assim, a arquitetura proposta nesta pesquisa foca no aspecto de realizar alterações rápidas e dinâmicas para Segunda Tela. Proporcionando a um ADST ser mutável sem a necessidade de instalar outras versões para atualizar suas funcionalidades.

Os usuários precisam ter apenas um ADST instalado em seu dispositivo móvel, com acesso à *internet*. E, quando houver atualização de novas emissoras ou funções de programações de emissoras, elas serão atualizadas diretamente em um único *app*. Com isso, os usuários não precisam encher seus dispositivos de aplicativos de Segunda Tela de diversas emissoras.

A atualização dos dados para as funcionalidades do aplicativo se dará por meio de tecnologias *web* usando a *internet*. Dentre elas, um *web service* denominado Atualizador de Serviços responsável pelo sincronismo das atualizações dos pacotes de

dados enviados pelas emissoras ao ADST. Esses pacotes podem ser imagens, textos, vídeos, *url* entre outros.

A arquitetura foi desenvolvida para ser executada na *internet*, não existindo necessidade de equipamentos adicionais para os usuários do ADST, sendo necessário apenas: TV e dispositivo móvel (*smartphone* ou *tablet*) com conexão à *internet*. Para infraestrutura de funcionamento dos serviços, serão necessários servidores de banco de dados e para hospedagem dos serviços *web*, mas que não são de responsabilidade do usuário do ADST.

4.2 Arquitetura Proposta

É fundamental para um ADST a entrega de atualizações das informações das emissoras em tempo quase real, visto que os dados transmitidos podem mudar em horários imprevisíveis daquilo que é transmitido na Primeira Tela. Assim, fatores como o tempo de atualização é visto como um complicador para o Atualizador de Serviços do ADST. Para tanto, a fim de garantir a entrega das informações atualizadas entre emissoras e ADST, foi implementada a solução com o serviço *push notifications*.

O serviço *push notifications* é ativado sempre que houver inclusão, alteração ou exclusão de informações para o aplicativo. As *push notifications* se comunicam com aplicativos instalados nos dispositivos móveis, podendo enviar texto, arquivos de mídia ou dados específicos do aplicativo, tais como sons de alerta ou sinais para exibir no ícone do aplicativo (YUAN, SCIENTIST e HEAL, 2012).

“Um push notification é uma mensagem ou alerta entregue por um servidor centralizado a um dispositivo final. Em outras palavras, a notificação push, também chamada de notificação de envio do servidor, é a entrega de informações de uma aplicação de software para um dispositivo de computação sem um pedido específico do cliente.”⁴

⁴ Extraído de: <http://www.devmedia.com.br/parse-com-como-criar-notificacoes-push-com-android/33171>

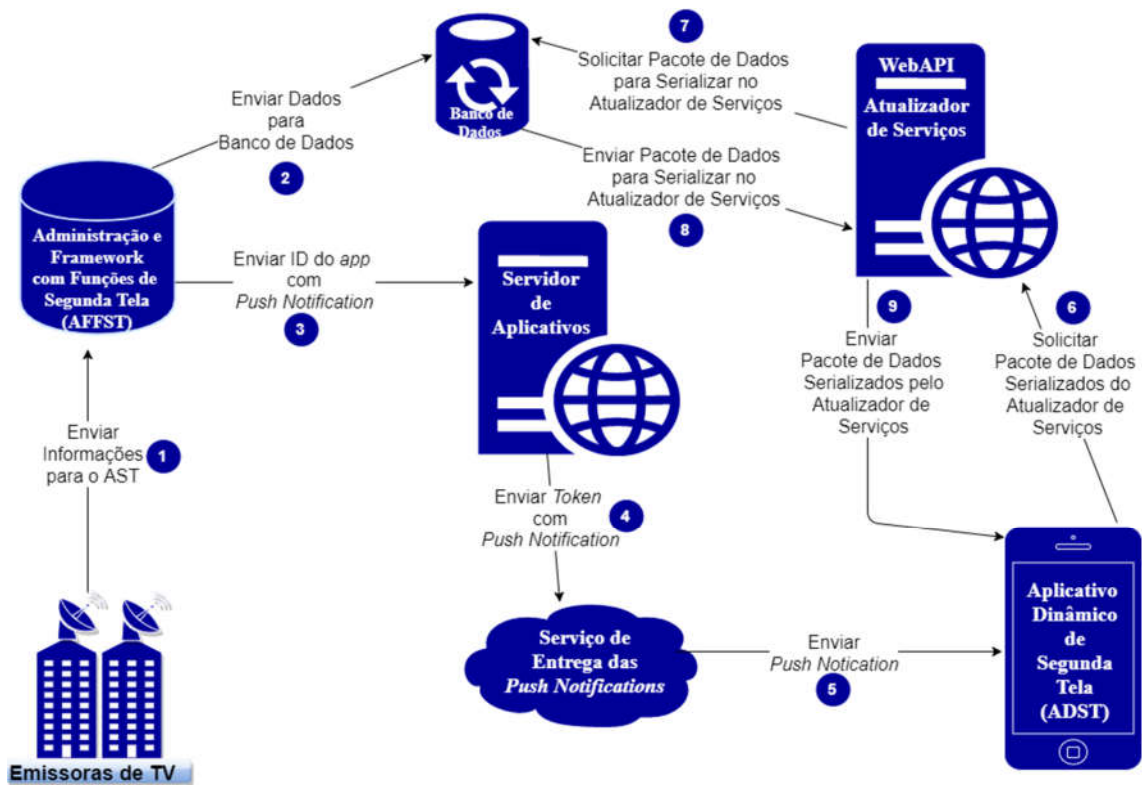
As *push notifications* chegam até os aplicativos através de um identificador “ID” que identifica o *token* do *app* nos servidores das lojas de aplicativos. O *token* é único e localiza o dispositivo *versus* aplicação. O que possibilita encontrar em quais dispositivos o aplicativo está instalado, sendo usado pelo serviço de entrega *push notifications* no envio das notificações (WARREN, MEADS, *et al.*, 2014).

Em Almeida, Abreu, *et al.* (2015) foram realizados os testes no recebimento de notificações exibidas em dispositivos móveis, TV ou em ambos, onde as notificações eram recebidas respectivamente em intervalos de 10, 30 e 60 segundos. A finalidade era informar a existência de informações da TV no aplicativo. No entanto, as notificações atrapalhavam ou retiravam a atenção dos usuários pelo fato deles as receberem em primeiro plano nos *apps* ou diretamente na tela da TV.

Para que as notificações não distraiam os telespectadores no momento que chegam no ADST, serão recebidas e executadas em segundo plano, ou também chamado de modo *background*, a fim de iniciar a atualização sem a intervenção do usuário. O modo de execução *background* é recomendado em Apple, Developer (2016) para implementações de tarefas de longa duração como o caso de aplicativos *mobile* que precisam baixar e processar novos conteúdos regularmente.

No recebimento das notificações pelo ADST, classes e objetos da programação iniciam métodos que solicitam dados ao Atualizador de Serviços. As *push notifications* notificam o ADST logo após serem realizadas inclusões, alterações ou exclusões de dados no Bando de Dados, como mostrado na Figura 4.1 e detalhado o processo na Tabela 4.1 que apresentam o funcionamento geral da arquitetura.

Figura 4.1 - Arquitetura de atualização dinâmica ao ADST



Fonte: Autor (2017)

Tabela 4.1 - Etapas gerais da arquitetura dinâmica

#	Descrição
1	As emissoras realizam o cadastro das informações que precisam ser enviadas ao ADST.
2	Os dados são armazenados no Banco de Dados a partir do cadastro feito pelo AFFST. No momento em que é realizado o cadastro pelo AFFST é confirmado sua inclusão, alteração ou exclusão no Banco de Dados.
3	Com a confirmação do cadastro no Banco de Dados, métodos de programação na aplicação AFFST solicitam ao Servidor de Aplicativos o envio do <i>push notifications</i> aos dispositivos móveis com o ADST. Na notificação é informado qual dado deve ser atualizado no ADST. Servidores de aplicativos como, por exemplo: <i>Google</i> , <i>Apple</i> ou <i>Microsoft</i> recebem o ID do aplicativo. Após o recebimento do ID, o servidor de aplicativos localiza o <i>token</i> que contém as informações do aplicativo e em quais dispositivos o <i>app</i> se encontra instalado.

4	O Servidor de aplicativos envia os <i>tokens</i> ao Serviço de Entrega das <i>push notifications</i> para que envie as notificações aos dispositivos com o ADST.
5	As notificações são recebidas pelo ADST informando que existe algum tipo atualização. Essas atualizações podem ser: nova emissora, novo programa, horários dos programas, menu, imagens ou outros. Tais informações devem chegar ao ADST por meio dos seus identificadores (Ids) dos dados armazenados nas tabelas do Banco de Dados.
6	Após o recebimento da notificação <i>push notifications</i> no ADST, um método de programação no ADST solicita ao Atualizador de Serviços o Pacote de Dados com os dados serializados.
7	O Atualizador de Serviços recebe a solicitação do ADST e, posteriormente solicita os dados ao Banco de Dados para serem serializados.
8	O Atualizador de Serviços recebe os dados do Banco de Dados e cria um Pacote de Dados serializado para ser enviado ao ADST.
9	O Atualizador de Serviços envia o Pacote de Dados serializado ao ADST por um arquivo no formato <i>JavaScript Object Notation</i> (JSON), informando o que deve ser incluído, alterado ou excluído. Finalmente, o ADST ao receber o Pacote de Dados serializado, são iniciados métodos de programação no ADST para que o Pacote de Dados seja deserializado e consumido pelo ADST.

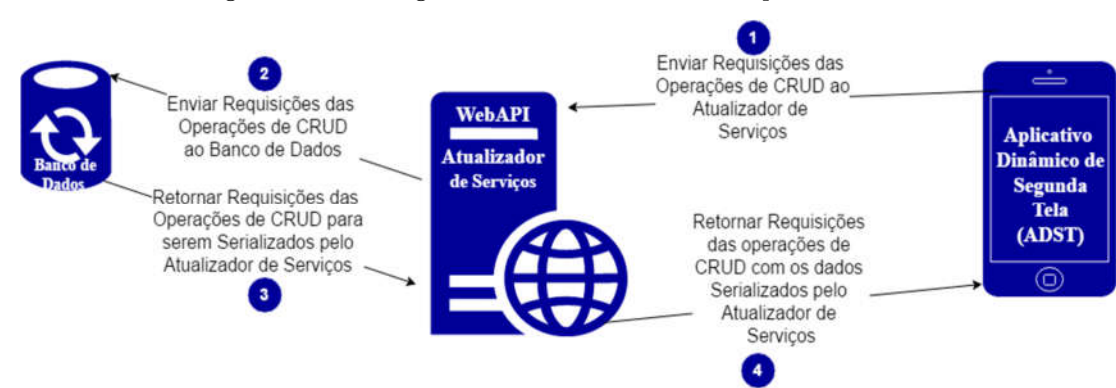
Fonte: Autor (2017)

A atualização dinâmica ocorre pela arquitetura, principalmente pelo do Atualizador de Serviços, um Web API capaz de enviar os dados serializados ao *app* através de arquivos JSON. O *app* recebe a *string* enviada no arquivo JSON, e o ADST desserializa a *string* do arquivo JSON e a transforma em um objeto que é consumido e posteriormente renderizado o resultado no ADST.

Operações de *Create*, *Read*, *Update* e *Delete* (CRUD) também são realizadas no ADST, como por exemplo consultar horários de programação das emissoras, fazer inserção de dados, curtir uma página, comentar sobre um programa, realizar compras. As operações de CRUD serão realizadas por arquivos JSON podendo ser com os métodos POST, GET, PUT ou DELETE através do HTTP, como citados em trabalhos como (BARNARD e PIERRO, 2014) e (NIU, YANG e ZHANG, 2014).

Na Figura 4.2 mostra o cenário dos processos de CRUD do ADST juntamente com o ambiente de serviços do Atualizador de Serviços. O processo sempre será iniciado no momento que o ADST enviar dados de requisições das operações de CRUD ao Atualizador de Serviços, ao qual deve realizar o envio das requisições das operações de CRUD ao Banco de Dados. Por fim, é confirmado ao ADST a inclusão, alteração, exclusão ou consulta das requisições. A descrição mais detalhada das etapas do processo é mostrada na Tabela 4.2.

Figura 4.2 - Dados gravados no Banco de Dados a partir do ADST



Fonte: Autor (2017)

Tabela 4.2 - Etapas do processo das operações de CRUD iniciadas pelo ADST ao Atualizador de Serviços

#	Descrição
1	O ADST envia requisições de CRUD serializadas no formato JSON para o Atualizador de Serviços.
2	O Atualizador de Serviços recebe as requisições de CRUD do ADST, e posteriormente as envia para o Banco de Dados onde serão consumidas.
3	O Banco de Dados recebe e realiza as operações de CRUD do ADST enviadas pelo Atualizador de Serviços.
4	O Atualizador de Serviços recebe a confirmação das operações de CRUD e as serializa enviando no formato JSON para o ADST. Finalmente ao ADST receber o arquivo JSON com os dados serializados, são iniciados métodos de programação no ADST para que os dados sejam deserializados e consumidos.

Fonte: Autor (2017)

4.3 Atualizador de Serviços

O Atualizador de Serviços deve garantir que os Pacotes de Dados cheguem ao ADST em tempo suficiente para atender ao horário que a programação da Primeira Tela estiver sendo exibida. Com o Atualizador de Serviços, são realizadas as atualizações do ADST, onde as *push notifications* resolvem parcialmente o problema do tempo de atualização.

Dessa forma, para garantir que os Pacotes de Dados sejam entregues em tempo suficiente das programações, foi proposto a solução *Mock servers* do *software* Postman⁵, por possuir uma API que se integra com o Atualizador de Serviços, ao qual pode ser usado para qualquer sistema integrado via HTTP ou HTTPS como discutido em (BLOOM, 2017). Utilizado na validação das requisições e respostas de *web services*, para realizar testes de desempenho, segurança e funcionalidade. Existem ainda outros *softwares* que podem ser usados, são eles: JMeter, SoapUI, Wcfstorm, Parasoft, Runscope, IBM Rational Service Tester for SOA Quality, WebInject, Postman e Ranorex.

O Postman, com a solução Mock Server, deve testar o envio das requisições HTTP para serviços RESTful pelo Atualizador de Serviços, enviando parâmetros e recebendo as respostas. Podendo, assim, analisar o tempo gasto em cada requisição, facilitando o teste e depuração dos serviços executados pelo Atualizador de Serviços.

A cada requisição do HTTP, é verificado quanto tempo médio gasto para que os Pacotes de Dados cheguem ao ADST. A medição é feita pelo levando em consideração o tempo levado para alcançar a quantidade de ADST instalados nos dispositivos. As informações para se saber q quantidade de *apps* instalados são

⁵<https://www.getpostman.com/>

disponibilizadas pelo Google e Apple, através de APIs respectivamente encontradas no Google Play Developer API⁶ e no Apple Developer⁷.

Dessa forma, se o tempo levado para o Pacote de Dados chegar ao ADST for maior que o tempo da programação exibida, então o Atualizador de Serviços não envia o Pacote de Dados aos *apps* com ADST. Assim, os Pacotes de Dados para criar funcionalidades e conteúdos somente serão entregues ao aplicativos pelo Atualizador de Serviços:

- se a programação que estiver sendo exibida ainda estiver dentro do seu horário de transmissão;
- se o tempo médio de envio e resposta das requisiões calculada pelo Postman com a solução *Mock servers* tiver verificado se o tempo médio gasto para o Pacote de Dados chegar aos aplicativos for suficiente para alcançar o tempo da transmissão da programação na Primeira Tela.

Fatores como infraestrutura de rede, configurações de servidores de hospedagem e velocidade de internet, são relevantes, pois, influenciam no tempo de atualização na entrega dos dados.

4.4 Habilitando ADST para Receber Funções e Conteúdos

O Atualizador de Serviços envia o Pacote de Dados após a sua criação no AFFST, onde o ADST solicita ao Atualizador de Serviços o envio do Pacote de Dados, como no exemplo da Figura 4.3. Esse Pacote de Dados é responsável por montar o ambiente e habilitá-lo para receber os conteúdos das funcionalidades para Segunda Tela.

Figura 4.3 - Modelo do Pacote de Dados para habilitar ADST

⁶ <https://developers.google.com/android-publisher/>

⁷ <https://developer.apple.com/documentation/>

```

1 {
2   "emissora": 1,
3   "emissoraNome": "TV Globo",
4   "programaId": 1,
5   "programaNome": "Campeonato Brasileiro",
6   "funcaoId": 1,
7   "funcaoNome": "Enquete",
8   "campos": [
9     {
10      "tipoId": 1,
11      "tipoCampo": "Label",
12    },
13    {
14      "tipoId": 2,
15      "tipoCampo": "Checkbox",
16    },
17    {
18      "tipoId": 3,
19      "tipoCampo": "Button"
20    }
21  ]
22 }

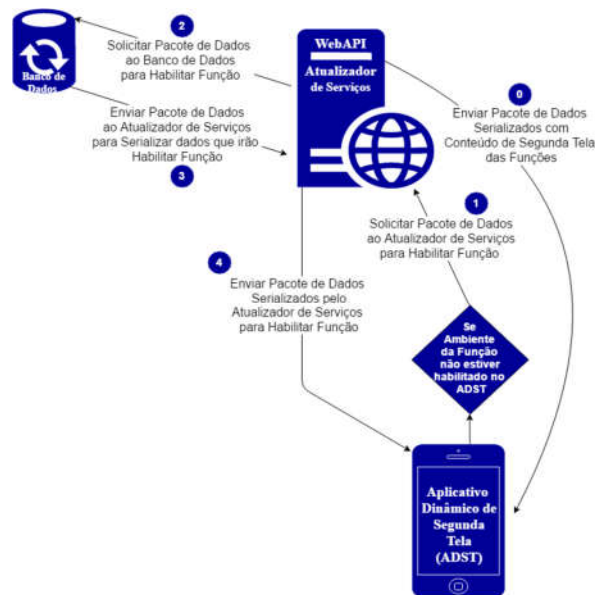
```

Fonte: Autor (2017)

No momento que o Pacote de Dados é deserializado no ADST, é informado ao aplicativo sobre qual é a função, quais tipos de campos ela deve possuir e qual programa da emissora será disponibilizado o conteúdo. Então, com a função habilitada, o ADST fica aguardando o conteúdo de Segunda Tela ser enviado.

As etapas que verificam e analisam quando deve ser habilitada as funções no ADST são mostradas na Figura 4.4 e detalhadas na Tabela 4.3.

Figura 4.4 - Etapas para habilitar funções no ADST



Fonte: Autor (2017)

Figura 4.5 - Pseudocódigo para habilitar ADST

```

se (FuncaoJaExiste) {
    execute StringJSON_DadosDaFuncao;
} senão se (FuncaoNaoExiste){
    execute StringJSON_CriarFuncao;
    execute StringJSON_DadosDaFuncao;
}

```

Fonte: Autor (2017)

Tabela 4.3 - Descrição das etapas para habilitar funções do ADST

#	Descrição
0	O ADST recebe o Pacote de Dados com conteúdo de Segunda Tela enviado pelo Atualizador de Serviços. Esse conteúdo será renderizado se o <i>app</i> já estiver com a função habilitada.
1	Se o ADST já estiver habilitado, poderá ser recebido o Pacote de Dados do conteúdo de Segunda Tela serializado pelo Atualizador de Serviços. Ou, caso contrário, como mostra o pseudocódigo da Figura 4.5, se o ADST não tiver com a função habilitada para receber conteúdos, deverá solicitar ao Atualizador de Serviços um Pacote de Dados, com informações para habilitar a função.
2	O Atualizador de Serviços solicita os dados ao Banco de Dados para serem serializados pelo Atualizador de Serviços.
3	O Banco de Dados envia os dados para que seja montado o Pacote de Dados serializado no Atualizador de Serviços.
4	O Atualizador de Serviços envia o Pacote de Dados serializado ao ADST, com todas as informações necessárias para montar e habilitar a função. No recebimento do Pacote de Dados pelo ADST ocorre a deserialização, e finalmente é habilitada a função.

Fonte: Autor (2017)

Com a função habilitada, o ADST está pronto para receber os conteúdos para Segunda Tela, que serão enviados pelas emissoras ao ADST por meio de um Pacote de Dados mostrado no exemplo na Figura 4.6. Nesse Pacote de Dados são enviadas todas as informações para que o conteúdo seja recebido pelo ADST. Por fim, após o recebimento do conteúdo da função, quando o telespectador abrir o aplicativo a interface estará renderizada dinamicamente, conforme apresentado na Figura 4.7.

Figura 4.6 - Modelo de string JSON com conteúdo de Segunda Tela

```
1 {  
2   "emissora": 1,  
3   "programaId": 1,  
4   "dataDeExibicao": "2017-04-16T00:00:00",  
5   "horaInicio": "2017-04-16T13:00:00",  
6   "horaFim": "2017-04-16T15:00:00",  
7   "funcao": [  
8     {  
9       "funcaoId": 1,  
10      "tipoIdHabilitar": [1,2,3],  
11      "perguntaLabel": "O Brasil é o seu time do coração?"  
12    }  
13  ]  
14 }
```

Fonte: Autor (2017)

Figura 4.7 - Tela da Enquete do ADST renderizada dinamicamente



Fonte: Autor (2017)

4.5 Padrão para Transferência e Comunicação

O padrão adotado na transferência e comunicação dos dados entre Atualizador de Serviços e ADST será o protocolo de comunicação *Representational State Transfer* (REST), baseado no protocolo *Hypertext Transfer Protocol* (HTTP) tendo os recursos

identificados pela *Uniform Resource Identifier* (URI) como citados em (NIU, YANG e ZHANG, 2014). Considerado muito eficaz no compartilhamento de dados na *internet*, como mostrado na pesquisa de (WO e SOO, 2015).

Grandes empresas como *Google*, *Facebook*, *LinkedIn* e *Netflix* possuem *Web APIs* baseadas no protocolo de comunicação REST, ao qual os serviços são acessados por diversos clientes tanto na requisição como na resposta.

*“...Web API é uma estrutura que torna mais fácil para construir serviços HTTP que chegam a uma ampla gama de clientes, incluindo navegadores e dispositivos móveis. ASP.NET Web API é uma plataforma ideal para a criação de aplicativos RESTful no .NET Framework.”*⁸

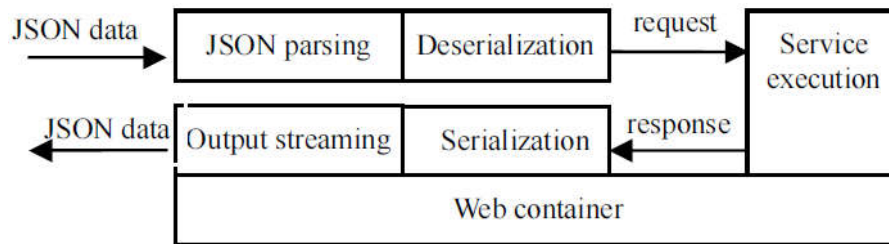
Para as requisições e respostas da arquitetura do ADST com o Atualizador de Serviços o desenvolvedor pode escolher o formato mais adequado para as mensagens do sistema, sendo os mais usados são: JSON e Extensible Markup Language (XML). Para tanto, foi escolhido o formato JSON, por ser fácil de testar, de se manter e de se consumir com *web service*. E por ser um formato livre, fortemente utilizado por outras aplicações como *Google*, *Facebook*, *LinkedIn* e *Netflix*, e por proporcionar o uso de diversos recursos utilizados pelo protocolo HTTP.

O XML diferentemente do JSON, possui redundância de dados excessiva e o processo de análise de dados é complexo, citado em (NIU, YANG e ZHANG, 2014) que apresentam comparações entre JSON e XML, e ainda afirmam que o JSON possui alto desempenho para requisições e respostas de dados em aplicações de serviços web.

Em Peng, Cao e Xu (2011) é mostrado a escalabilidade usando JSON na serialização e deserialização das requisições e respostas de dados, onde é provado que JSON tem melhor desempenho do que usando XML. E ainda, é mostrado cinco etapas para o procedimento de invocação de requisição e resposta a *web service* baseado em JSON como mostrado na Figura 4.8 e descritas na Tabela 4.4.

⁸ Extraído de: <http://www.asp.net/web-api>

Figura 4.8 - Etapas para Invocar *Web Service* baseado em JSON



Fonte: Peng, Cao e Xu (2011)

Tabela 4.4 - Etapas para o procedimento de invocação de requisição e resposta ao *web service*

#	Etapas
1	Analisar dados JSON, o que significa que o servidor analisará os dados assim que receber os dados de solicitação dos clientes.
2	Os dados JSON analisados são deserializados em dados de aplicação.
3	O servidor invoca o <i>Web Service</i> e captura os resultados. O tempo que leva é relevante para a complexidade computacional do serviço web.
4	São serializados os resultados retornados de dados da aplicação em dados JSON.
5	A etapa final que grava os dados JSON em um fluxo de saída de HTTP e, em seguida, são enviados para o lado dos clientes.

Fonte: Autor (2017)

As requisições e respostas da comunicação usam elementos específicos na construção do REST podendo ser encontrados em Aece (2013), as quais são descritos na Tabela 4.5.

Tabela 4.5 - Elementos usados na construção do REST. Fonte Aece (2013)

Elementos	Descrição
Recursos	Podem ser qualquer coisa que esteja disponível e deseja-se acessar. Pode-se entender o recurso como uma página HTML, produtos de uma loja, um vídeo, uma imagem e também uma funcionalidade que pode ser acessível através de outras aplicações.

URI	Todo e qualquer recurso para ser acessível necessita de um mecanismo de identificação para que possa efetivamente ser encontrado e manipulado. Nesse sentido a web provê o padrão <i>URI</i> que determina o endereço onde ele está localizado identificando unicamente um <i>recurso</i> ao mesmo tempo em que o torna endereçável e manipulável por um protocolo de aplicação, como o HTTP.
Representação	É uma visão de um estado do <i>recurso</i> em determinado instante do tempo, esta visão é codificada em formatos padronizados e passíveis de transferência através da web, como, XHTML, JSON, XML, CSV e etc.
Media Type	O recurso retornado ao cliente é uma fotografia em um determinado momento, seguindo um determinado formato. O <i>media type</i> é responsável por determinar o formato em que a requisição é enviada e como a resposta é devolvida. Alguns <i>media types</i> : imagens, textos, vídeos, etc.

Fonte: Autor (2017)

Na comunicação, métodos de programação são executados no AFFST, iniciando o processo de transferência dos dados ao ADST, enviando notificações *push notifications* para informar a existência das funcionalidades no mesmo instante em que os dados são gravados no Banco de Dados. Por sua vez, o ADST, ao receber as notificações, realiza chamadas a métodos de programação, solicitando ao Atualizador de Serviços o envio dos Pacotes de Dados serializados para que possa receber, deserializar e transformar em objetos para serem consumidos.

Capítulo 5

5 Protótipo

5.1 Aplicativo Dinâmico de Segunda Tela (ADST)

O Aplicativo Dinâmico de Segunda Tela (ADST) é a aplicação da arquitetura proposta para Segunda Tela, onde emissoras de televisão oferecem aos telespectadores um novo conceito de interatividade através funções de Segunda Tela que complementam a programação durante a programação. Quanto a usabilidade, o aplicativo oferece navegação de simples utilização para que usuários menos experientes com tecnologia facilmente compreendam e manuseiem, proporcionando maior engajamento com a interface do aplicativo.

Quando o aplicativo é aberto, na tela inicial é exibida uma lista com todas as emissoras, onde o telespectador pode escolher a emissora. Caso o telespectador não encontre facilmente a emissora, vai estar disponível a opção: *Localizar emissora*, conforme mostrado na Figura 5.1.

Figura 5.1 - Tela inicial para selecionar emissora no ADST



Fonte: Autor (2017)

Após escolher a emissora as informações e funções de Segunda Tela do programa são exibidas como mostrado na Figura 5.2. São apresentadas as funções de Segunda Tela que foram disponibilizadas pela emissora pra aquele programa, as quais foram enviadas até o *Aplicativo de Segunda Tela* pelo *Atualizador de Serviços*.

Figura 5.2 - Tela com informações do programa e funções de Segunda Tela no ADST

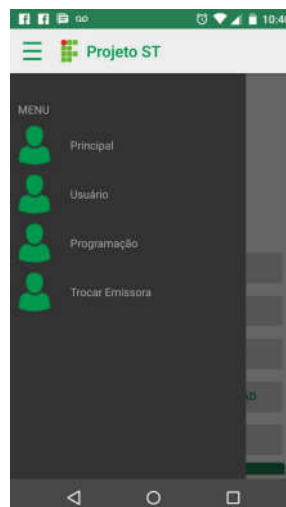


Fonte: Autor (2017)

O menu do ADST ilustrado na Figura 5.3 é disponibilizado no *app* somente após o telespectador escolher uma emissora, suas opções estão descritas abaixo:

- Principal: faz com que o telespectador volte para a tela inicial do *app*.
- Usuário: o telespectador poderá criar seu perfil, com isso terá acesso a mais opções no ADST, por exemplo: *participar de comentários, curtir programas e comprar produtos*.
- Programação: o telespectador tem acesso a relação da *grade de programação da emissora*, onde ele pode abrir programas que não estão sendo exibidos e ter acesso opções como: *Sobre* e *Curtir*, para as demais funções são disponibilizadas somente durante o horário do programa.
- Trocar emissora: durante a experiência de Segunda Tela em programas de uma emissora específica o usuário pode sentir a necessidade de acessar outros programas em emissoras diferentes. Para isso o ADST disponibiliza no menu a opção *trocar emissora* proporcionando ao telespectador mudar de emissora e emergir outros programas que possuam funções de Segunda Tela.

Figura 5.3 - Tela do menu do o ADST



Fonte: Autor (2017)

5.2 Administração de Funções para Segunda Tela (AFFST)

O AFFST é o administrador de conteúdo do ADST, é um ambiente que recebe as informações dos cadastros das emissoras, das programações, dos dias, dos horários em que os programas serão exibidos e outros como mostrado na Figura 5.4. Nele são cadastrados os fusos horários das localidades onde as emissoras estão localizadas geograficamente, para que sejam respeitadas as diferenças de fuso horário de cada uma delas.

Figura 5.4 - Tela de cadastro das emissoras no AFFST



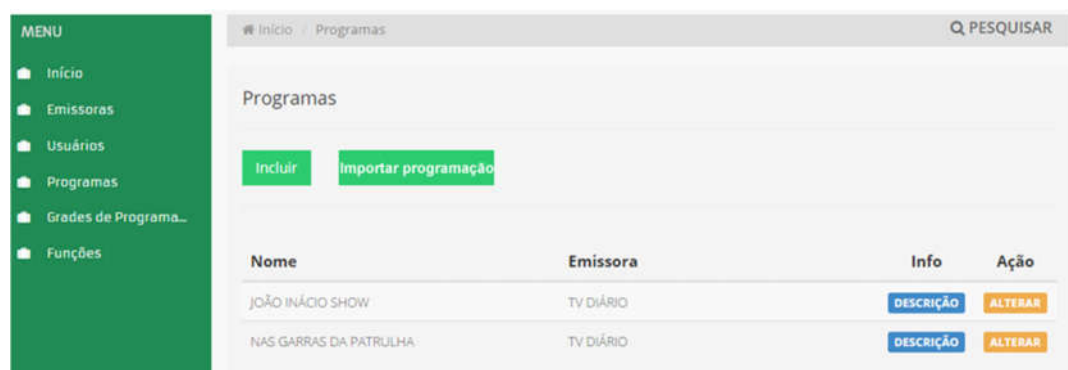
Fonte: Autor (2017)

Com as emissoras cadastradas no AFFST, será possível importar a *grade de programação* que pode ser exportado pela emissora através de integrações de outros softwares com o AFFST. Caso a emissora não tenha disponibilidade para exportar o arquivo com sua grade de programação, poderá cadastrar manualmente diretamente no ambiente do AFFST conforme mostrado na Figura 5.5.

A importação das informações dos programas das emissoras com os dias e horários se dará através do arquivo no formato XML enviados pelas emissoras afim de serem importados no AFFST. O arquivo XML conterá tags com: nome da emissora;

programas da emissora; grade de programação com dias e horários que os programas são exibidos. As emissoras deveram seguir esse padrão para a criação do arquivo XML, para que sejam importadas suas grades de programação no AFFST. A importação no AFFST ocorre utilizando o botão *Importar programação* que se encontra no menu *Programas* mostrado na Figura 5.5, onde são importados a programação com dias e horários que serão exibidos na Primeira Tela.

Figura 5.5 - Tela de cadastro dos programas das emissoras no AFFST



Fonte: Autor (2017)

Já com todos os programas da emissora cadastrados ou importados, é possível informar ou alterar no AFFST o período que os programas serão exibidos, a partir da opção *Grades de Programação*. Nele é informado quais os horários dos programas da emissora mostrado na Figura 5.6. As emissoras que importaram a programação pelo AFFST utilizando o XML, suas informações da grade de programação já estarão importadas, com isso não é necessário realizar novamente o cadastro, no entanto, podem realizar alterações nos dados já importados.

Figura 5.6 - Tela de cadastro da grade de programação das emissoras no AFFST

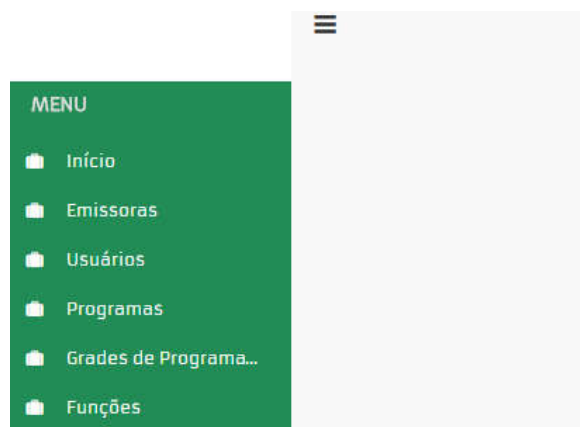


Nome	Emissora	Dt. exibição	Início	Fim	Ação
JOÃO INÁCIO SHOW	TV DIÁRIO	17/05/2016	17/05/2016 08:13	17/05/2016 12:30	ALTERAR
NAS GARRAS DA PATRULHA	TV DIÁRIO	10/04/2016	04/04/2016 00:25	07/04/2016 23:59	ALTERAR

Fonte: Autor (2017)

As emissoras cadastradas no AFFST terão o domínio de toda sua programação e de suas funções de Segunda Tela. Por isso, o controle de acesso no ambiente do AFFST será por usuário de emissoras, no entanto, haverá um usuário administrador que terá o controle sobre todas as emissoras cadastradas, esse usuário será o administrador do AFFST. Sendo assim o usuário da emissora terá controle somente das programações e funções do seu domínio. O usuário administrador poderá ter o controle sobre todos os domínios das emissoras e de todas as opções disponíveis no AFFST, como mostradas na Figura 5.7.

Figura 5.7 - Tela do menu AFFST



Fonte: Autor (2017)

O AFFST disponibilizará uma *biblioteca de funções*, disponível para que as emissoras adicionem funções aos programas, como no exemplo mostrado na Figura 5.8 onde mostra a emissora criando a função *Enquete* para um programa da sua grade programação. Essas funções serão disponibilizadas no ambiente e posteriormente serão sincronizados pelo *Atualizador de Serviços* até o *Aplicativo Dinâmica de Segunda Tela*.

Figura 5.8 - Tela de cadastro das funções, exemplo da criação da função Enquete para programa da emissora no AFFST

A interface de cadastro de funções no sistema AFFST apresenta uma barra lateral verde com o menu e uma área principal de formulário. O formulário, intitulado 'Função', possui os seguintes campos:

- Programa*:** Dropdown menu com a opção 'JOÃO INÁCIO SHOW' selecionada.
- Grade de Programação*:** Dropdown menu com a opção selecionada exibindo detalhes: 'JOÃO INÁCIO SHOW | Dt. Exibição: 17/05/2016 | Início: 17/05/2016 08:13 | Fim: 17/05/2016 12:30'.
- Tipo*:** Dropdown menu com a opção 'Enquete' selecionada.
- Pergunta*:** Campo de texto para a pergunta da enquete.

Um botão verde com o texto 'Concluir' está localizado no canto inferior direito do formulário.

Fonte: Autor (2017)

As emissoras criam funções seguindo padrão para Segunda Tela, sendo sincronizadas pelo *Atualizador de Serviços*. As funções podem ser criadas sem que a emissora necessite de conhecimento técnico de programação.

As funções não devem romper características de Segunda Tela, tendo em vista que o *app* oferece conteúdo para complementar os programas da Primeira Tela, afim de enriquecer o conteúdo através da Segunda Tela.

As funcionalidades criadas pelo AFFST permitem que o aplicativo ofereça ao telespectador funções de Segunda Tela, sendo essas sincronizadas através do *Atualizador de Serviços*, que tem um papel fundamental de distribuir as funcionalidades até os aplicativos, com isso é atendido o item 18 (dezoito) do *checklist*.

Das funções que serão apresentadas para o ADST, duas delas sempre vão estar disponíveis, são as funções: *Sobre o programa* e *Curtir*, elas aparecerão para todos os programas que forem criados a partir do AFFST, ou seja, mesmo que as emissoras não habilitem outras funções para os programas, mesmo assim será disponibilizado para o telespectador essas duas funções que serão chamadas de funções padrão.

Das funções de Segunda Tela que o AFFST disponibilizará ao ADST, serão apresentadas funções de monetização de aplicativos de Segunda Tela citadas em (LIMA e LIMA, 2015), são maneiras que podem ser exploradas para monetizar o aplicativo. As principais formas de monetização são:

- Exibição de anúncios: Podem ser cobrados através do custo por mil, como na televisão, e neste caso ainda permitem um maior controle sobre o número de visualizações.
- Publicidade Interativa: Os dispositivos de Segunda Tela permitem que o usuário interaja com a publicidade veiculada, assim como na Internet, proporcionando uma experiência mais engajadora com a marca que está anunciando, sem interromper o fluxo da programação na tela da TV.
- *Mobile Commerce (m-commerce)*: Venda de produtos e serviços para o público da aplicação; A opção *M-Commerce* inicia sempre quando houver produtos para venda no que estiver sendo transmitido na Primeira Tela, as compras em Segunda Tela podem ser exploradas principalmente nos intervalos da programação.
- Anúncios em vídeo: Similar ao adotado pelo *youtube* pode ser inserido nos intervalos da programação ou em *pre-roll*, antes do usuário acessar algum conteúdo.

Essas formas de monetizações são atraentes para emissoras de televisão, tendo em vista que são formas comerciais de atrair recursos financeiros e inovar o modelo de

negócios das emissoras de televisão, buscando aumentar seus clientes e parceiros com: Publicidade e Propaganda, *Merchandising* e *M-Commerce*.

O ADST vai dispor de opções de vídeos para programações como novelas, filmes, seriados, jogos esportivos, reality, jornais, shows e programas de auditório, oferecendo opções de assistir vídeos de *programas que já ocorreram*, *vídeos na íntegra* e *making-of*.

Para programações como jogos esportivos e programas de auditório, que por muitas vezes realizam pesquisas com os usuários, faz com que os telespectadores participem decidindo por meio de escolhas do que deve acontecer. Ou sendo consultados sobre algum tipo de tema, para isso serão disponibilizadas duas funções chamadas de: *Quizz* e *Enquete*.

No *ADST* será disponibilizado funções que devem ajudar questões como regionalismo, termos técnicos, gírias, ou palavras de difíceis compreensão da população. Ou situações que necessitem de detalhes sobre determinada pessoa, animal, ferramental, espécie ou qualquer outro tipo de coisa, a função será chamada de *Glossário*.

No *Aplicativo de Segunda Tela* os telespectadores podem *comentar*, *curtir*, *compartilhar*, *adicionar imagens no perfil*, *interagir com personagens* (elenco) e *produtores* (autores e diretores). O ambiente possui funcionalidades de redes sociais, no entanto o telespectador terá todas essas funções no ambiente do *ADST*, que poderão ser compartilhadas para redes sociais.

As *Funções* de Segunda Tela do AFFST, dará opções para as emissoras criarem novas funções, o dinamismo proporcionará criar novas funções a partir de uma biblioteca já definida, com ele as emissoras diminuem a necessidade de conhecimento técnico de programação.

Capítulo 6

6 Estudo de Caso

Afim de validar a arquitetura de atualização dinâmica para Segunda Tela, serão considerados quando e em quanto tempo se dará a criação e entrega dos dados ao ADST pela arquitetura. Será apresentado não só a forma do funcionamento da arquitetura, como também a forma como o telespectador será deparado com a Segunda Tela, tendo em vista que o telespectador espera que as funções enviadas ao ADST estejam sempre disponíveis com a programação, seja ela ao vivo ou gravada.

6.1 Cenário para Validação da Arquitetura

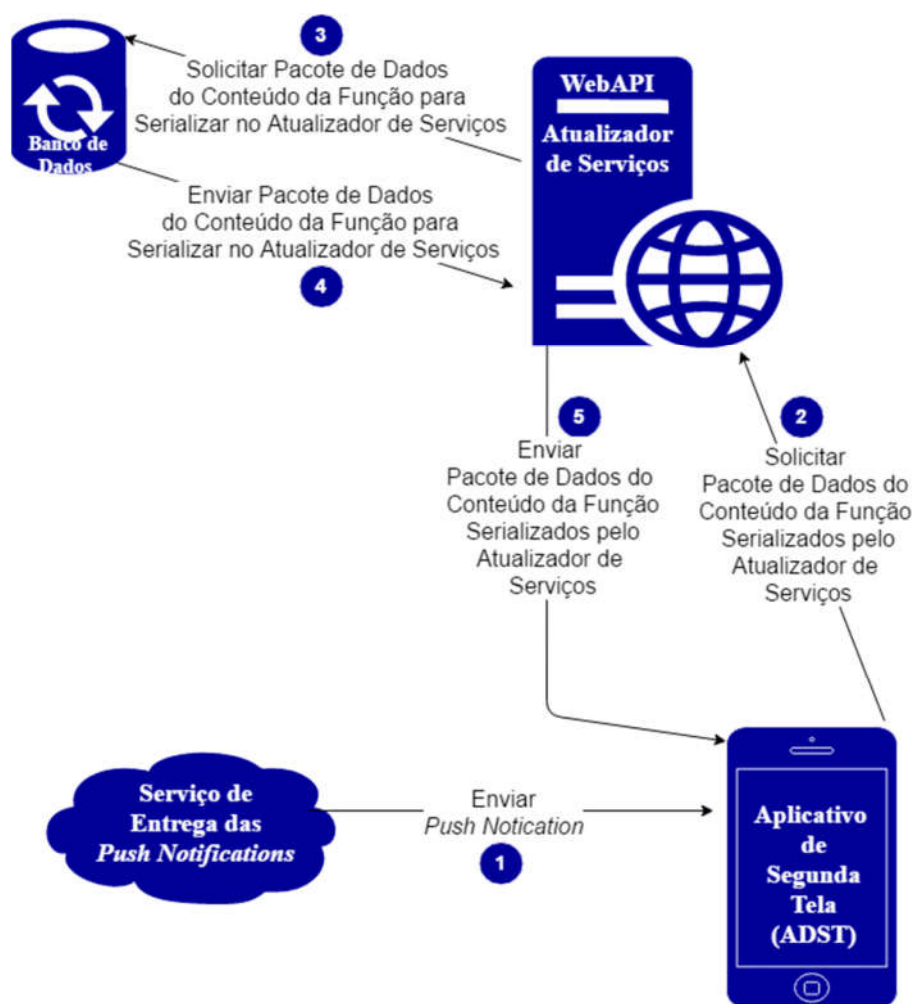
Em uma transmissão de partida de futebol que tem a duração média de 90 minutos divididos em 02 tempos, o ADST receberá o conteúdo das funções de Segunda Tela obedecendo ao tempo em que o programa estiver sendo transmitido. Durante a transmissão da partida de futebol, algumas funções com seus conteúdos já foram previstas e enviadas ao ADST. Porém, podem ser criadas funções e conteúdos durante a transmissão da partida de futebol, das quais não tinham sido previstas.

Para esse cenário, percebeu-se durante a transmissão da partida de futebol a necessidade de ser criado o conteúdo da função Enquete. A função Enquete no ADST deverá ser informada pelo narrador da partida, e estar disponível aos telespectadores

durante os 15 minutos após o início da partida do 1º tempo. O tempo que os telespectadores terão para responder a Enquete será de 30 minutos.

A função deve estar disponível no ADST antes do narrador anunciar aos telespectadores. Para tanto, será necessário que emissora cadastre o conteúdo da função pelo AFFST. Após o cadastro, é enviada a *push notifications* ao ADST notificando da criação da Enquete, como mostrado na Figura 6.1 e descrito da Tabela 6.1.

Figura 6.1 - Etapas para atualização da função Enquete no ADST



Fonte Autor (2017)

Tabela 6.1 - Elementos usados na construção do REST

#	Descrição
1	Envio da <i>push notification</i> ao ADST para informar que existe um novo conteúdo
2	O ADST solicita ao Atualizador de Serviços o Pacote de Dados serializado com o conteúdo da função Enquete
3	O Atualizador de Serviços solicita ao Banco de Dados os dados para preenchimento do Pacote de Dados do conteúdo da função Enquete
4	O Banco de Dados retorna ao Atualizador de Serviços o Pacote de Dados para a criação do conteúdo da função Enquete
5	O Atualizador de Serviços com e a API da solução <i>Mock servers</i> do Postman, somente enviará o Pacote de Dados serializado para a criação do conteúdo da função Enquete ao ADST, se: • a programação estiver sendo exibida, • ou, se o tempo que a API da solução <i>Mock servers</i> do Postman levou para validar a entrega do Pacote de Dados for suficiente para alcançar a programação que está sendo exibida na Primeira Tela

Fonte: Aece (2013)

6.1.1 Validar o Tempo da Entrega do Pacote de Dados da Função

Enquete

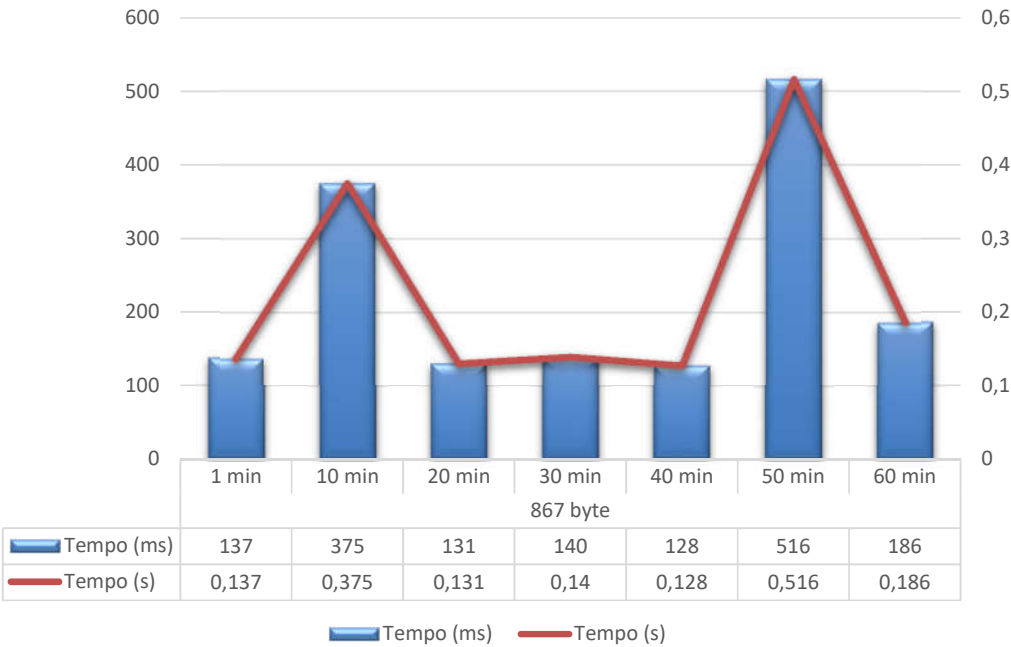
Para realizar a validação de envio e recebimento dos pacotes de dados do cenário da criação da função Enquete, usou-se: um servidor da Amazon para hospedar o AFFST e os serviços web do Atualizador de Serviços, com as seguintes configurações: processador Xeon 3.00 Ghz, memória física de 32 GB, *internet* banda larga dedicada de 50 MB; um computador pessoal equipado com processador i7 2.00 GHz, memória física de 16 GB e *internet* banda larga de 5 MB.

Com a validação usando o *software* Postman e o Atualizador de Serviços, foi calculado o tempo médio de comunicação para o Atualizador de Serviços enviar e receber o Pacote de Dados serializado. A mesma rotina realizada nos testes que serão apresentados é feita internamente pelo Atualizador de Serviços com a API da solução *Mock servers* do Postman.

O teste foi realizado com um Pacote de Dados do tamanho de 867 bytes e foram realizados em intervalos intercalados de 10 minutos durante o período de 1 hora para que se pudesse analisar ao final o tempo médio utilizado para receber e enviar o Pacote de Dados ao ADST

Foi possível analisar o tempo médio que o Atualizador de Serviços levou para receber e enviar os dados serializados ao ADST. O tempo médio foi de 230,43 milissegundos (ms) ou 0,23 segundos (s) a partir dos valores levantados apresentados na Figura 6.2.

Figura 6.2 - Análise do recebimento e envio do Pacote de Dados no Atualizador de Serviços para criação da Enquete no ADST



Fonte: Autor (2017)

Observou-se que o ADST recebe o Pacote de Dados em um tempo médio satisfatório. Para o cenário da partida de futebol, supõe-se que aos 15 minutos de partida o narrador já deve ter feito todas as etapas do envio da Enquete para que o telespectador ao abrir o ADST encontre-a disponível.

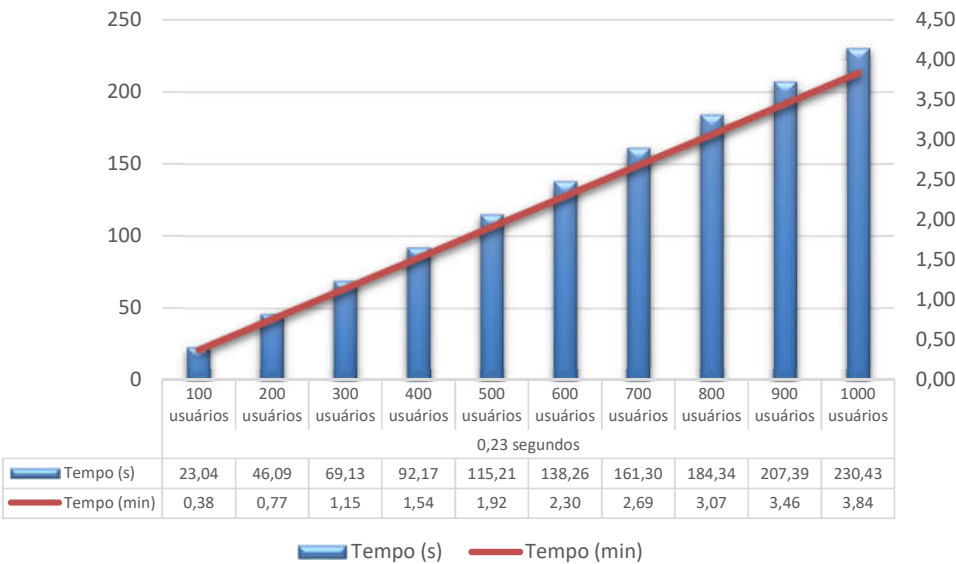
Os processos de envio deverão acontecer antes dos 15 minutos, mas, deve-se levar em consideração um número de pessoas a ser alcançada para que se saiba quantos minutos antes dos 15 minutos a função vai estar no ADST.

Então, para validar a entrega do Pacote de Dados a um determinado número de usuários, realizou-se outra análise a partir do resultado do tempo médio de 0,23 segundos (s). Nessa análise buscou-se alcançar o número de 1000 usuários, sendo realizado um levantamento da análise do tempo a cada 100 usuários.

Verificou-se que o tempo gasto para alcançar os 1000 usuários, foi aumentando progressivamente. No resultado apresentado na Figura 6.3 para o recebimento da função Enquete para os 1000 usuários do ADST, se fez necessário o tempo médio de 3,84 minutos (min).

Para que todos os usuários recebam o conteúdo de Segunda Tela antes do narrador da partida informar que a função está disponível, a função deverá ser enviada pela arquitetura dinâmica aos 3,84 minutos (min) antes dos 15 minutos da partida.

Figura 6.3 - Análise do recebimento e envio do Pacote de Dados no Atualizador de Serviços para criação da Enquete no ADST



Fonte: Autor (2017)

Nesse cenário da partida de futebol observou-se que o Pacote de Dados e a infraestrutura são fatores relevantes, para o tempo levado para alcançar um determinado número de usuários. Contudo, chegamos na informação do tempo necessário para que se pudesse alcançar uma quantidade específica de usuários. Tornando a emissora capaz de enviar dados de Segunda Tela para seus programas ao vivo ou gravados, em tempo compatível com o horário que será realizada a transmissão.

Capítulo 7

7 Conclusão

7.1 Contribuição

Este trabalho apresentou uma arquitetura para aplicativos de Segunda Tela capaz de serem atualizados dinamicamente. A arquitetura resolve o dinamismo em aplicações de Segunda Tela em relação ao seu desenvolvimento tradicional através do Atualizador de Serviços.

Buscou-se mostrar a eficiência na troca de dados usando computação baseada em serviços. Adotou-se neste trabalho o JSON como o formato de troca de dados para aplicações de Segunda Tela, com uma abordagem eficiente para serializar e deserializar os dados trocados, apresentando-se relevante no desempenho de serviços *web*, e com as notificações em tempo real utilizando o serviço *push notifications*.

A arquitetura dinâmica para Segunda Tela apresentada neste trabalho, possibilita ter seus dados e interface atualizados dinamicamente, alcançando o tempo de entrega dos Pacotes de Dados de atualizações de Segunda Tela para programas ao vivo ou gravados, por meio um Atualizador de Serviços.

7.2 Trabalhos Futuros

Assim, estudos futuros poderão utilizar a metodologia aplicada para aplicações *mobile* de uso geral que necessitam ter seus dados e interfaces atualizados dinamicamente.

A arquitetura proposta pode ser usada em aplicações que precisam ser atualizadas dinamicamente em tempos programados. Espera-se que em trabalhos futuros sejam contempladas outras pesquisas para aplicativos que precisam ser atualizados sem a necessidade de ser atualizado a partir das lojas de aplicativos.

As tecnologias que foram utilizadas precisam de mais pesquisas que possibilitem um tempo menor para a atualização dinâmica das informações na aplicação de Segunda Tela a fim de otimizar a arquitetura proposta nesta pesquisa e melhorar o seu desempenho.

7.3 Publicação

Architecture Support for Second Screen Apps Dynamic Adaptation.
Webmedia 2017 – XXIII Simpósio Brasileiro de Sistemas Multimídia e Web. Outubro 17–20, 2017, Gramado, Brasil. © 2017 Association for Computing Machinery. ACM ISBN 978-1-4503-5096-9/17/10...\$15.00.
<https://doi.org/10.1145/3126858.3126890>.

Referências

AECE, I. Microsoft Developer Network. **ASP.NET Web API - HTTP, REST e o ASP.NET**, jul. 2013. Disponível em: <<https://msdn.microsoft.com/pt-br/library/dn369238.aspx>>. Acesso em: 27 set. 2016.

ALMEIDA, P. et al. Notification Mechanisms In Second-Screen Scenarios Towards a Balanced User Experience. **2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN)**, Turin, 2015. p. 95-99. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7325492&isnumber=7325470>>. Acesso em: 01 dez. 2016.

ANGELUCI, A. C. B. Recomendações de IHC para uso de aplicativos interativos em televisão e segunda tela a partir de infraestrutura de TVD. **Tese de Doutorado, Escola Politécnica, Universidade de São Paulo**, São Paulo, 14 out. 2013. Disponível em: <www.teses.usp.br>. Acesso em: 01 fev. 2017.

APPLE, DEVELOPER. Guides and Sample Code. **App Programming Guide for iOS - Background Execution**, 13 set. 2016. Disponível em: <<https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html>>. Acesso em: 02 dez. 2016.

ARNOLD, M.; CHEN, X.-M.; BAUM, P. A phase-based audio watermarking system robust to acoustic path propagation, vol 9, nº 3. **IEEE Transactions on Information Forensics and Security**, Germany, 2014. p. 411-425. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6678326&isnumber=6727454>>. Acesso em: 22 set. 2014.

ASHMORE, S.; MAKKI, S. K. IMISSAR: an intelligent, mobile middleware solution for secure automatic reconfiguration of applications, utilizing a feature model approach. **In Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication (ICUIMC '11)**,

New York, NY, USA, n. 58, 2011. p. 1-7. Disponível em:

<<http://dx.doi.org/10.1145/1968613.1968684>>. Acesso em: 06 out. 2016.

ASKWITH, I. Television 2.0 – Reconceptualizing Tv as an engagement medium.

Master of Science in Comparative Media Studies. **Comparative Media Studies**, 10 out. 2007. Disponível em: <<http://cmsw.mit.edu/television-2-0-tv-as-an-engagement-medium/>>. Acesso em: 1 set. 2016.

BARNARD, B.; PIERRO, M. D. Extending Collection+JSON and Automatic CRUD Interfaces. **2014 IEEE 17th International Conference on Computational Science and Engineering**, Chengdu, 19-21 dez. 2014. p. 1692-1696. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7023822&isnumber=7023510>>. Acesso em: 15 dez. 2016.

BISIGNANO, M.; DI MODICA, G.; TOMARCHIO, O. Dynamic user interface adaptation for mobile computing devices. **2005 Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops)**, 2005. p. 158-161.

BLOOM, J. D. MockServer - Easy mocking of any system you integrate with via HTTP or HTTPS. **Mock Server**, 2017. Disponível em: <<http://www.mock-server.com/>>. Acesso em: 01 jun. 2017.

BRAGANÇA, S. L. D. Introdução à análise estatística de variáveis dicotômicas e aplicações em dados socioeconômicos. **Revista Brasileira de Economia**.

Fundação Getúlio Vargas / EPGE, Rio de Janeiro, v. 31, n. 2, jun. 1977. p.

323-352. Disponível em:

<<http://bibliotecadigital.fgv.br/ojs/index.php/rbe/article/view/192>>. Acesso em: 10 set. 2016.

CALIXTO, G. M.; ANGELUCI, A. C. B.; KURASHIMA, C. S. Effectiveness analysis of audio watermark tags for IPTV second screen applications and synchronization.

Telecommunications Symposium (ITS), 2014 International, São Paulo, 17-20 ago. 2014. 1-5. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6947989&isnumber=6947948>>. Acesso em: 2 set. 2016.

CASTILLEJO, E.; ALMEIDA, A.; LÓPEZ-DE-IPÍÑA, D. Modelling users, context and devices for adaptive user interface systems. **International Journal of**

Pervasive Computing and Communications, 2014. 69-91. Disponível em:

<<http://dx.doi.org/10.1108/IJPCC-09-2013-0028>>.

CHAPUIS, O.; BEZERIANOS, A.; FRANTZESKAKIS, S. Smarties: An Input System for Wall Display Development. **In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)**, New York, NY, USA, 2014. p. 2763-2772. Disponível em:

<<http://doi.acm.org/10.1145/2556288.2556956>>. Acesso em: 11 out. 2016.

CHEN, G. et al. A Framework for Practical Dynamic Software Updating. **IEEE Transactions on Parallel and Distributed Systems**, 27, 01 abr. 2016. p. 941-950. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7103360&isnumber=7430414>>.

CHMIELEWSKI, J. et al. Declarative GUI descriptions for device-independent applications. **Personal Ubiquitous Comput.**, London, UK, UK, abril 2016. p. 185-194. Disponível em: <<http://dx.doi.org/10.1007/s00779-016-0903-2>>.

CORTEZ, J.; SHAMMA, D. A.; CAI, L. Device communication: a multi-modal communication platform for internet connected televisions. **In Proceedings of the 10th European conference on Interactive tv and video (EuroITV '12)**.

ACM, New York, NY, USA, 2012. p. 19-26. Disponível em:

<<http://dl.acm.org/citation.cfm?id=2325622&dl=ACM&coll=DL&CFID=837832734&CFTOKEN=83117035>>. Acesso em: 07 ago. 2016.

GEERTS, D. et al. In front of and behind the second screen: viewer and producer perspectives on a companion app. **Proceedings of the 2014 ACM international conference on Interactive experiences for TV and online video**, New York, NY, USA, 2602312, 2014. p. 95-102. Disponível em:

<<http://doi.acm.org/10.1145/2602299.2602312>>.

HICKS, ; NETTLES,. Dynamic Software Updating. **ACM Trans. Program. Lang. Syst.**, New York, 27, nov. 2005. 1049--1096. Disponível em:

<<http://doi.acm.org/10.1145/1108970.1108971>>.

HOWSON, C. et al. Second screen TV synchronization. **2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)**, Berlin, 2011. p. 361-365. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6031815&isnumber=6031793>>.

ILVONEN, V.; IHANTOLA, P.; MIKKONEN, T. Dynamic Software Updating Techniques in Practice and Educator's Guides: A Review. **2016 IEEE 29th**

International Conference on Software Engineering Education and Training (CSEET), Dallas, 2016. 86-90. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7474470&isnumber=7474423>>. Acesso em: 01 mar. 2017.

KRONE, M.; MÜLLER, C.; ERTL, T. Remote Rendering and User Interaction on Mobile Devices for Scientific Visualization. **In Proceedings of the 8th**

International Symposium on Visual Information Communication and

Interaction (VINCI '15), New York, NY, USA, 2015. p. 21--26. Disponível em:

<<http://dx.doi.org/10.1145/2801040.2801057>>. Acesso em: 10 out. 2016.

LIMA, M. M. D.; LIMA, F. D. S. Aplicativos de Segunda Tela: Participação, Interatividade e Modelo de Negócios através de Dispositivos Móveis. **Comunicação e Mercado - Revista Internacional de Ciências Sociais Aplicadas da**

UNIGRAN, Mato Grosso do Sul, v. 4, n. 10, dez. 2015. p. 19-30. Disponível em:

<<http://www.unigran.br/mercado/paginas/arquivos/edicoes/10/2.pdf>>. Acesso em: 2 set. 2016.

LIRA, S. A.; NETO, A. C. Coeficientes de correlação para variáveis ordinais e

dicotômicas derivados do coeficiente linear de Pearson. **Revista Ciência &**

Engenharia. Universidade Federal de Uberlândia - UFU, Recife, v. 15, n. 1/2,

dez. 2006. p. 45-53. Disponível em:

<<http://www.seer.ufu.br/index.php/cieng/article/view/529>>. Acesso em: 11 set. 2016.

LOUBERRY, C.; ROOSE, P.; DALMAU, M. Kalimucho: Adaptation Platform for Mobile Applications. **2011 11th Annual International Conference on New**

Technologies of Distributed Systems, Paris, 2011. p. 1-8. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5957977&isnumber=5957970>>. Acesso em: 02 mar. 2017.

MURARASU, A. F.; MAGEDANZ, T. Mobile Middleware Solution for Automatic Reconfiguration of Applications. **Information Technology: New Generations,**

2009. ITNG '09. Sixth International Conference on, Las Vegas, NV, 2009. p.

1049-1055. Disponível em:

<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5070762&isnumber=5070575>>. Acesso em: 06 out. 2016.

MURRAY, J. H. **Hamlet no Holodeck - O futuro da narrativa no ciberespaço**. São Paulo: Fundação Editora UNESP, 2003.

NAHUZ, T. P.; SANTOS, M. C. D. Assistindo a Televisão: O Papel Auxiliar dos Aplicativos de Segunda Tela para o Storytelling Transmídia. **Simpósio Internacional de Tecnologia e Narrativas Digitais. Encontro de Grupos e Redes de Pesquisas em Comunicação e Tecnologia I Rodada de Articulação de Projetos Internacionais de Pesquisa**, São Luís, MA, Brasil, 18 nov. 2015. Disponível em: <<http://www.labcomdata.com.br/wp-content/uploads/2015/12/NahuzTPPaper.pdf>>. Acesso em: 14 ago. 2016.

NIU, ; YANG, ; ZHANG,. A design of cross-terminal web system based on JSON and REST. **2014 IEEE 5th International Conference on Software Engineering and Service Science**, Beijing, junho 2014. p. 904-907. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6933711&isnumber=6933501>>.

PENG, D.; CAO, L.; XU, W. Using JSON for Data Exchanging in Web Service Applications. **Journal of Computational Information Systems**, 2011. p. 5883-5890. Disponível em: <https://www.researchgate.net/publication/265874991_Using_JSON_for_Data_Exchanging_in_Web_Service_Applications>.

PUNT, M. et al. An integrated environment and development framework for social gaming using mobile devices, digital TV and Internet. **Multimedia Tools Appl**, Hingham, MA, USA, 18 set. 2015. p. 8137-8169. Disponível em: <[10.1007/s11042-014-2045-8](https://doi.org/10.1007/s11042-014-2045-8)>. Acesso em: 08 set. 2016.

RODRIGUES, C. H. A.; REIS, E. D. L.; LUCENA, V. F. D. Second Screen prototype for broadcasted digital TV users in ISDB-Tb standard. **2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)**, Tokyo, 7-10 out. 2014. p. 266-267. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7031342&isnumber=7031081>>. Acesso em: 08 ago. 2016.

RUBIO, F. G.; ELENA, F. B.; DAVIC, A. G. Analysis of uses and interaction with second screen. **CUADERNOS.INFO**, 2014. p. 159-175. Disponível em: <<http://cuadernos.info/index.php/CDI/article/view/cdi.35.565>>. Acesso em: 05 jul. 2016.

SANTOS, M. C. D.; NAHUZ, T. Aplicativos de Segunda Tela: Mapeamento de Utilização e Desenvolvimento de Ferramenta de Autoria para Produtores de Conteúdo. **Intercom – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação. XXXVIII Congresso Brasileiro de Ciências da**

Comunicação, Rio de Janeiro, RJ, 4 a 7 set. 2015. p. 1-15. Disponível em:
<<http://portalintercom.org.br/anais/nacional2015/resumos/R10-1425-2.pdf>>. Acesso em: 1 set. 2016.

SEGUNDO, R. M. C.; SANTOS, C. A. S. Systematic Review of Multiple Contents Synchronization in Interactive Television Scenario. **ISRN Communications and Networking**, 19 mar. 2014. Disponível em:
<<http://dx.doi.org/10.1155/2014/127142>>. Acesso em: 01 set. 2016.

SERUFFO, M. C. R. et al. Second Screen Application – Dengue Fever: Prevention through Information. **ACM International Conference on Interactive Experiences for Television and Online Video (TVX 2015)**, Bruxelas, Bélgica, 5 jun. 2015. Disponível em:
<https://figshare.com/articles/DEMO_103_Second_Screen_Application_Dengue_Fever_Prevention_through_Information/1423370>. Acesso em: 13 set. 2016.

ŠOŠKIĆ, N. et al. Second Screen User Experience: A New Digital Television Frontier. **Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on**, Opatija, 26-30 maio 2014. p. 1057-1060. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6859725&isnumber=6859515>>. Acesso em: 22 abr. 2016.

SOSKIC, N.; TAPAVICKI, D. P.; RAPIC, D. Remote control android application for second screen DTV environment. **2014 3rd Mediterranean Conference on Embedded Computing (MECO)**, Budva, Montenegro, 15-19 jun. 2014. p. 272-274. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6862714&isnumber=6862649>>.

SUTTER, R. D. et al. MediaCRM: enabling customer relationship management in the broadcast. **Proceedings of the 2011 International Conference on Web Information Systems and Mining - Volume Part I (WISM'11)**, Berlin, Heidelberg, 2011. p. 19-26. Disponível em:
<<http://dl.acm.org/citation.cfm?id=2045561.2045565>>. Acesso em: 28 set. 2016.

WARREN, I. et al. Push Notification Mechanisms for Pervasive Smartphone Applications. **IEEE Pervasive Computing**, 20 maio 2014. p. 61-71. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6818504&isnumber=6818495>>. Acesso em: 29 nov. 2016.

- WO, S. O.; SOO, H. K. Study on Access Permission Control for the Web of Things. **2015 17th International Conference on Advanced Communication Technology (ICACT)**, Seoul, 01 jan. 2015. p. 574-580. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7224926&isnumber=7224736>. Acesso em: 11 dez. 2016.
- YUAN, M. J.; SCIENTIST, C.; HEAL, R. IBM developerWorks. **Enviando notificações de push a dispositivos iOS**, 15 mar. 2012. Disponível em: <https://www.ibm.com/developerworks/br/library/mo-ios-push/>. Acesso em: 02 nov. 2016.
- ZHENG, M.; ORMANDJIEVA, O.; FAN, H. Designing Context Sensitive Mobile User Interface. **Proceedings of the International Conference on Software Engineering Research and Practice (SERP)**, 2015. 152.
- ZIEGLER, C. Second screen for HbbTV — Automatic application launch and app-to-app communication enabling novel TV programme related second-screen scenarios. **2013 IEEE Third International Conference on Consumer Electronics in Berlin (ICCE-Berlin)**, Berlin, 9-11 set. 2013. p. 1-5. Disponível em: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6697990&isnumber=6697949>. Acesso em: 01 set. 2016.

Anexo A - *Checklist* de Funcionalidades em *Apps* de Segunda Tela

1. Apresenta a opção de assistir ao conteúdo televisivo na íntegra?
2. Apresenta guias de episódios, resumos, biografia dos personagens, reportagens?
3. Apresenta fragmentos de episódios, capítulos ou programas anteriores? (cenas, reportagens, quadros?)
4. Apresenta histórias oficiais paralelas ao conteúdo primário? Ou seja, histórias que não são transmitidas com o conteúdo primário? (Conteúdo primário refere-se aos episódios ou capítulos da ficção)
5. Apresenta conteúdos trazidos do universo ficcional para o mundo real, como blogs ou sites?
6. Apresenta informações do mundo real que ajudam a entender o ficcional (enciclopédico)?
7. Apresenta informações sobre os bastidores? (Comentários oficiais do elenco ou equipe de produção sobre episódios)
8. Apresenta informações sobre aqueles que estão envolvidos com a criação do programa? (Elenco e equipe de produção)
9. Apresenta produtos para compra?
10. Apresenta conteúdo para download?
11. Apresenta jogos de conhecimento (quizz)?
12. Apresenta conteúdos para personalização pelo usuário (texto, imagem, áudio, vídeo), ou envio de conteúdo personalizado?

13. Apresenta espaço para que o telespectador possa inserir comentários, compartilhar e curtir conteúdo?
14. Apresenta interação entre usuários?
15. Apresenta interação entre usuários e produtores?
16. Apresenta interação entre usuários e personagens?
17. Apresenta plataformas de rede social?
18. Apresenta conteúdo sincronizado ao conteúdo primário?
19. Apresenta espaço onde a participação do usuário é reconhecida? (envio de comentários a serem veiculados na transmissão do conteúdo, participação em pesquisas e enquetes a serem veiculados...)
20. Apresenta a opção de interatividade onde a participação do usuário afeta o andamento do programa? (como votações comuns a reality shows ou escolha de finais alternativos)

Anexo B - *Apps* de TVs Brasileira e Internacional

B.1. *Apps* de TVs Brasileira e Internacional

No Brasil algumas emissoras de TV Aberta possuem aplicativos disponíveis nas lojas de app, esses aplicativos apresentam características de Segunda Tela. Dessa maneira, foi realizado um levantamento bibliográfico dos aplicativos das seguintes emissoras: Rede Globo de Televisão (TV Globo), Sistema Brasileiro de Televisão (SBT), Rede Record e Rede Bandeirantes (Band).

Inicialmente foram levantados os apps da Rede Globo⁹, conhecida no Brasil como TV Globo, essa emissora tem o sinal da sua programação distribuída em quase todo o território brasileiro, por meio de 5 emissoras próprias, em parceria com empresas afiliadas, e em mais de 100 países por meio da Globo Internacional. Disponibiliza várias

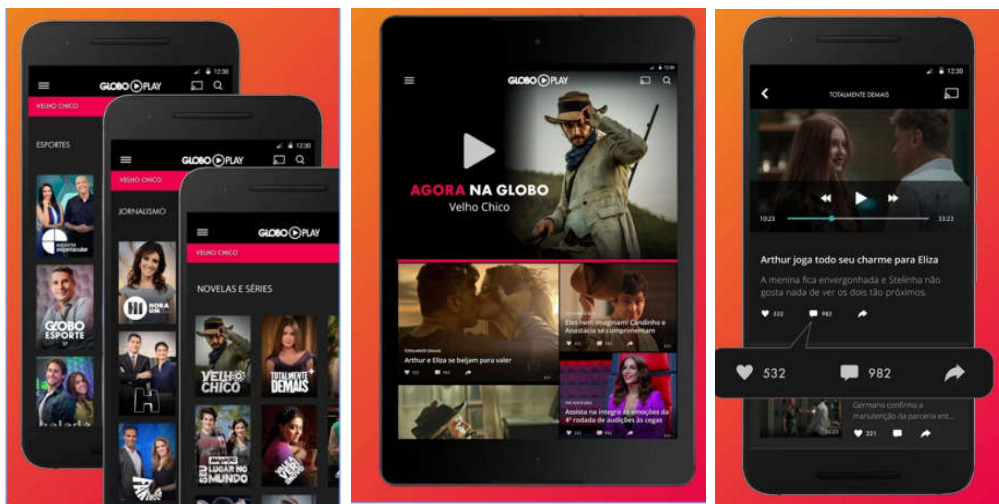
⁹ <http://grupoglobo.globo.com>

opções de aplicativos nas lojas de apps, desses, alguns possuem características significantes de Segunda Tela aos quais serão descritos a seguir.

O aplicativo Globo Play¹⁰ da emissora TV Globo mostrado na Figura B.1, oferece acesso gratuito a trechos de novelas, séries e minisséries, assim como a programas jornalísticos e telejornais esportivos. Outras opções são disponibilizadas para os assinantes, onde possuem acesso a íntegra de novelas, séries, programas de humor, câmeras exclusivas de reality quando houver, e acervo de programas jornalísticos, novelas, séries e minisséries.

Para a versão gratuita e para assinantes o Globo Play tem opção curtir e compartilhar em redes sociais os vídeos, além de comentários.

Figura B.1 - Telas de divulgação do aplicativo Globo Play



O Globo Play oferece *simulcasting* abreviação de *simultaneous broadcast*, em português, transmissão simultânea, ou melhor, a exibição simultânea da programação que está passando ao vivo na TV. Com isso é capaz de ter sincronismo do que é transmitido na Primeira Tela para a Segunda Tela.

¹⁰ <https://play.google.com/store/apps/details?id=com.globo.globotv>

No entanto, as transmissões ao vivo pelo Globo Play ainda não atendem todos os estados brasileiros, somente São Paulo e Rio de Janeiro em suas respectivas capitais e regiões metropolitanas, mas a emissora tem uma proposta de expansão para as cidades que tenham filiadas, para que sejam exibidos programas ao vivo da emissora Globo e também das programações ao vivo de suas filiadas conforme a localização em que se encontra o telespectador.

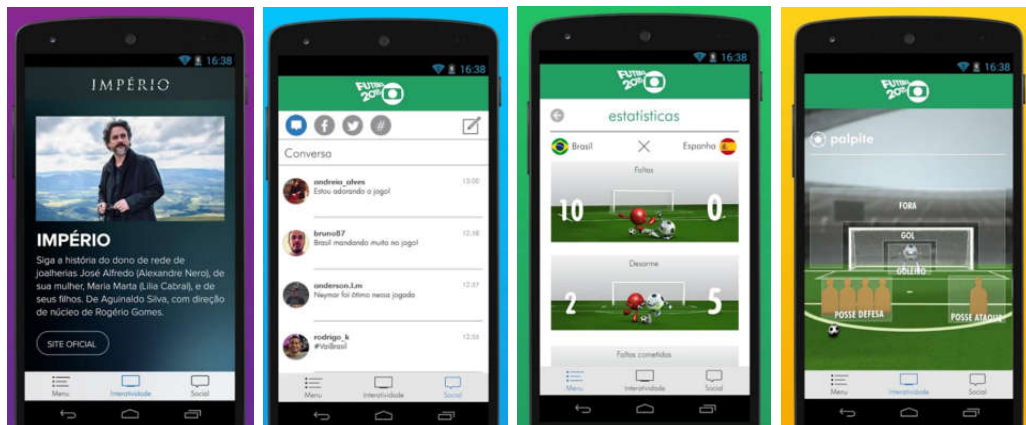
Para que o aplicativo determine a localização do telespectador, o app já dispõe da funcionalidade para localizar onde o telespectador encontra-se, através de GPS, e para que isso funcione é preciso habilitar o serviço de localização do GPS no dispositivo móvel.

Já no aplicativo Globo¹¹ mostrado na Figura B.2 traz informações com a programação de todos os programas que serão exibidos na emissora, com sinopses e imagens. A interatividade do aplicativo possibilita que o telespectador converse com outros sobre a transmissão da programação que está sendo exibida, o aplicativo possibilita que usuários comentem através de redes sociais mostrado também na Figura 2.7.

Quanto a transmissão de jogos de futebol o app da Globo oferece interatividade através de: games, replay, enquetes, quiz, placar, estatísticas e tudo sobre o jogo em tempo real mostrados também na Figura B.2.

¹¹ https://play.google.com/store/apps/details?id=com.deped.GloboMaisAndroid&hl=pt_BR

Figura B.2 - Telas de divulgação do aplicativo Globo



Por se tratar de uma emissora com filiadas espalhadas pelos estados brasileiros, em alguns momentos do dia elas tem sua programação própria, com isso, ocorrem em alguns momentos do dia o evento que está sendo apresentado no aplicativo é diferente do que está sendo exibido na TV em determinada região.

O app [Globoesporte.com](https://play.google.com/store/apps/details?id=com.globo.ge.app)¹² é um aplicativo de esporte e entretenimento que permite seus telespectadores terem acesso direto as notícias do seu clube, aos jogos em Tempo Real, e às principais notícias do futebol internacional e de outros esportes. Estão disponíveis ainda os vídeos dos jogos mais importantes e dos programas esportivos da Globo. Também ficam à disposição dos telespectadores as tabelas dos principais campeonatos do Brasil e do Mundo mostrados na Figura B.3. Além de enviar notificações sobre as principais notícias do esporte aos telespectadores mesmo que não esteja sendo transmitido nenhum evento esportivo na emissora.

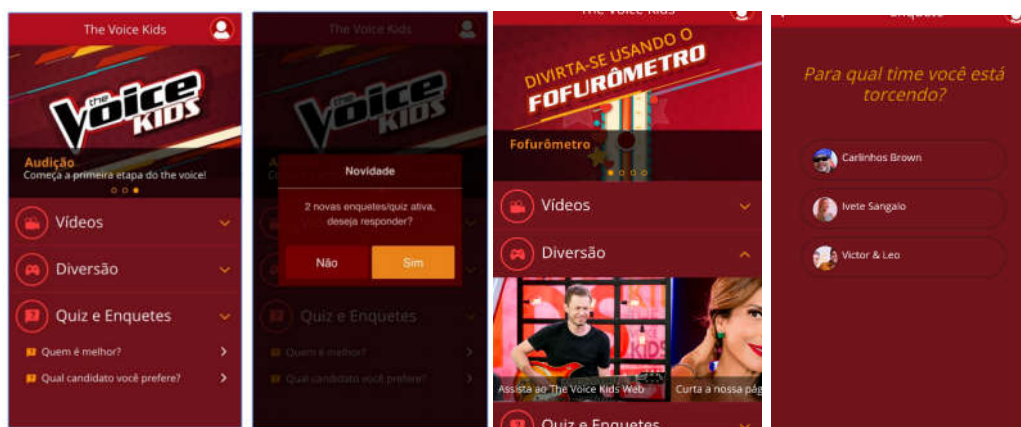
¹² <https://play.google.com/store/apps/details?id=com.globo.ge.app>

Figura B.3 - Telas de divulgação do aplicativo Globo



O aplicativo TheVoiceKids¹³ foi lançado na primeira edição brasileira do The Voice Kids reality show de competição musical destinado para crianças. O telespectador pode interagir com o programa pelo aplicativo participando de quizzes, enquetes, votação ao vivo, games, se divertir com os melhores vídeos e bastidores do programa mostrado na Figura B.4.

Figura B.4 - Telas retiradas do aplicativo TheVoiceKids



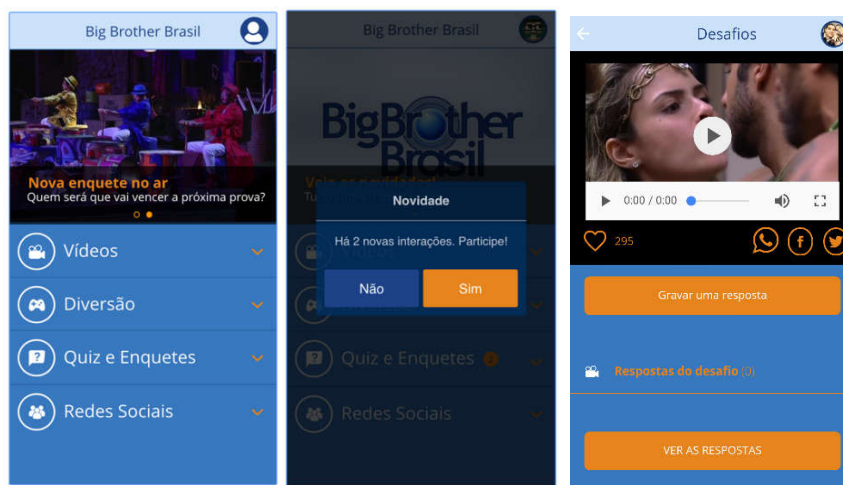
¹³ <https://play.google.com/store/apps/details?id=com.ionicframework.thevoicekids178077>

Entretanto esse aplicativo se restringe ao período em que o reality estiver sendo exibido, e após esse período cairá em desuso e somente terá funcionalidades quando houver uma nova edição do programa transmitido pela emissora.

O aplicativo BBB16¹⁴ se refere ao reality show chamado Big Brother Brasil da edição número 16 mostrado na Figura B.5. A partir do aplicativo o telespectador pode interagir com o programa através de votações, vídeos exclusivos, notícias, notificações de alertas sobre o que está acontecendo na casa, enquetes, quizzes, bate papo, vídeos desafios dublando cenas do programa.

Entretanto o aplicativo se restringe ao período em que o reality estiver sendo exibido, e após esse período cairá em desuso e somente terá funcionalidades quando houver uma nova edição do programa transmitido pela emissora.

Figura B.5 - Telas retiradas do aplicativo BBB



O Sistema Brasileiro de Televisão¹⁵ (SBT) é uma rede de televisão aberta brasileira, possui 114 emissoras, sendo 08 emissoras próprias, além de 107 emissoras afiliadas espalhadas por todo o território nacional. Atualmente, o SBT tem apenas um

¹⁴ <https://play.google.com/store/apps/details?id=com.ionicframework.bbb16262863>

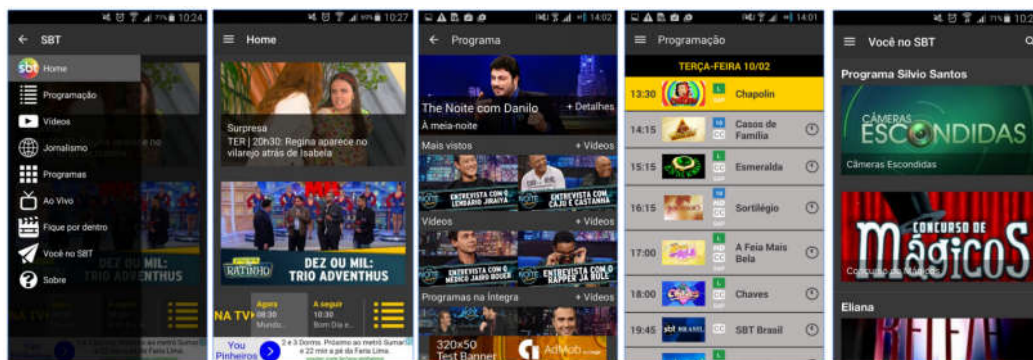
¹⁵ <http://www.sbt.com.br>

aplicativo nas lojas de apps, ao qual possui características significativas de Segunda Tela e será descrita a seguir.

O app SBT¹⁶ disponibiliza todo o catálogo de vídeos, inclusive conteúdos de algumas emissoras regionais, detalhes da grade de programação, lembretes de início do programa na TV, informações de bastidores dos programas e as últimas novidades.

Um dos diferenciais é a transmissão ao vivo do sinal, além de coberturas especiais, e também por oferecer serviços como alertas de programação e um vasto repositório de vídeos da programação que podem ser compartilhados em redes sociais externas. O aplicativo do SBT mostrado na Figura B.6 oferece ainda opções onde os telespectadores podem acompanhar postagens sobre personagens, apresentadores e tudo que envolve a programação da tela principal.

Figura B.6 - Telas retiradas do aplicativo SBT



A emissora de televisão aberta Rede Record¹⁷, é uma das maiores emissoras de televisão comercial do mundo. Tem uma abrangência de transmissão em quase todo território brasileiro, tem parcerias com várias emissoras filiadas espalhadas por todo território brasileiro. Nesse momento a Record possui apenas um aplicativo nas lojas de

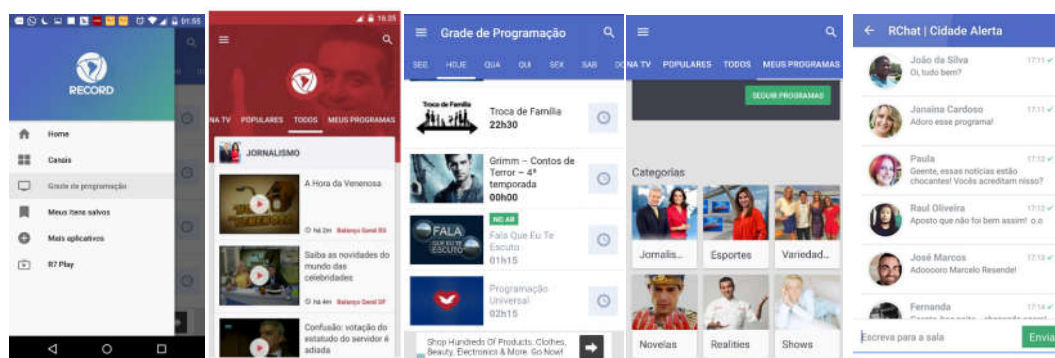
¹⁶ <https://play.google.com/store/apps/details?id=br.com.sbt.app>

¹⁷ <http://rederecord.r7.com>

apps, ao qual possui características significativas de Segunda Tela e será descrita a seguir.

No aplicativo Record¹⁸ pode ser acessada à grade completa da programação, com os melhores vídeos do jornalismo, esportes, variedades, novelas, linha de shows, também será possível guardar os vídeos na área de favoritos, compartilhar nas redes sociais, e acompanhar os mais vistos e populares mostrados na Figura B.7. Cada programa tem sua sala de bate-papo, que segue a grade de programação da emissora.

Figura B.7 - Telas retiradas do aplicativo Record



Ao final cita-se os apps da BAND¹⁹ emissora de televisão aberta brasileira, seu sinal é distribuído para todo o Brasil por meio das suas emissoras próprias ou afiliadas, televisão por assinatura e no exterior pela Band Internacional. Disponibiliza alguns aplicativos nas lojas de apps, desses, apenas um apresentou características significantes de Segunda Tela as quais serão descritos posteriormente.

O app BAND²⁰ possui mais de 20 horas diárias de transmissão ao vivo. Possibilita ao telespectador acompanhar o programa favorito diretamente no celular, pode conferir a programação, rever os vídeos que foram ao ar e obter informações

¹⁸ <https://play.google.com/store/apps/details?id=com.r7.rederecord>

¹⁹ <http://www.band.uol.com.br>

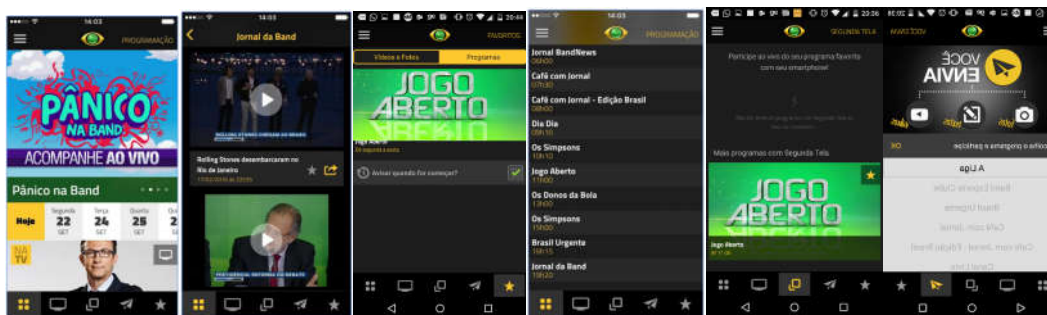
²⁰ <https://play.google.com/store/apps/details?id=br.com.band.guiatv>

complementares, sempre que a Segunda Tela estiver ativa mostrado na Figura B.8. O usuário possui uma área pessoal de favoritos para acessar facilmente tudo que mais gostar.

Foram destacados alguns recursos do menu do aplicativo, descritos abaixo:

- Band ao vivo: os principais programas ao vivo, diariamente, no seu celular;
- Grade de Programação: os horários de exibição da emissora. Essa área também serve de acesso aos vídeos de cada programa;
- Vídeos: esporte, notícia e entretenimento. Agora você pode rever o que passou na Band de forma gratuita, no seu celular, onde e quando quiser;
- Segunda Tela: nessa área você pode participar de enquetes, ter acesso a informações complementares sobre o que está passando na TV e ver fotos exclusivas dos bastidores. Você também pode conhecer outros fãs no bate-papo e conferir uma seleção de pérolas ditas por apresentadores, repórteres e celebridades. Tudo isso integrado às principais redes sociais para compartilhar com amigos;
- Você envia: nessa área pode ser enviado fotos, textos e vídeos a qualquer programa da emissora que são listados no app conforme imagem a seguir.

Figura B.8 - Telas retiradas do aplicativo SBT



As emissoras de televisão internacional com aplicativos em dispositivos móveis, investem em recursos para averiguar as potencialidades destes aplicativos e garantir ainda mais a credibilidade dos seus telespectadores. Os aplicativos que serão

apresentados a seguir, são das emissoras Cable News Network, *Fox Broadcasting Company*, Televisa e Entertainment and Sports Programming Network (ESPN).

A Cable News Network²¹ (CNN) é um canal a cabo norte-americana, um dos mais respeitados e confiáveis fontes de informações do mundo em oferecer notícias de última hora a análises profundas sobre economia, empresas e negócios, lazer, estilo de vida, esportes, atualidades, ciência e tecnologia. A CNN já oferece aos seus telespectadores aplicativo para Segunda Tela ao qual será descrito posteriormente.

No aplicativo CNN²² o telespectador tem alertas de notícias ao vivo, assistir CNN TV ao vivo, além de poder compartilhar em redes sociais, o app pode pedir a permissão para receber determinadas funções no seu dispositivo, tais como localização, marcar vídeos e matérias como favoritos, conexão com contas de redes sociais mostrados na Figura B.9. Também permite que o telespectador possa interagir sobre novas matérias, e configurar as informações que devem ser exibidas no *app* de acordo com a preferência determinada pelo usuário.

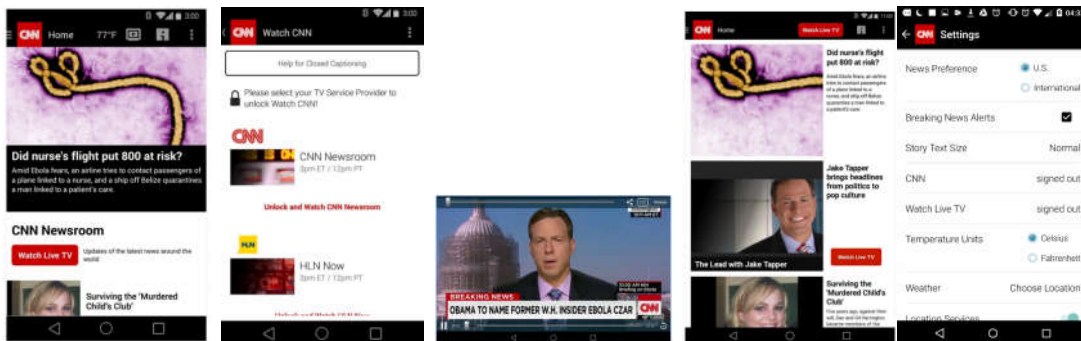
Algumas funções oferecidas pelo *app* CNN:

- Alertas de notícias;
- Clipes de vídeo;
- Cobertura de eventos ao vivo, porém para assistir precisa estar conectado a um serviço de TV;
- Informação do tempo;
- Galerias de fotos relacionadas com as notícias do dia.
- Sincronização com relógio do dispositivo móvel para receber as últimas notícias no topo.

²¹ <http://edition.cnn.com/>

²² <https://play.google.com/store/apps/details?id=com.cnn.mobile.android.phone>

Figura B.9 - Telas retiradas do aplicativo CNN



A emissora Fox Broadcasting Company²³ (Fox) é uma emissora de televisão com sede nos Estados Unidos. A emissora é um dos canais mais assistido por jovens e adultos, principalmente pelos seus programas que abordam assuntos parcialmente destinados a este público. Está presente em todo o mundo, sendo que em alguns países a programação da Fox não é a mesma da Fox americana. A emissora tem somente um aplicativo para Segunda Tela ao qual será descrito posteriormente.

Com o aplicativo FOX Play²⁴ o telespectador pode ter acesso ilimitado a clips, trailers, eventos de esportes, receitas, artesanato e os melhores momentos de seus programas favoritos, e conteúdos especiais e exclusivos da emissora mostrados na Figura B.10. O usuário do app poderá ter acesso a conteúdos completos dos canais FOX: séries, filmes, documentários, eventos esportivos ao vivo e muito mais, e conteúdo adicional.

A partir do app FOX Play o telespectador poderá:

- Criar sua própria lista com suas séries e programas favoritos;
- Continuar assistindo ao episódio do ponto onde parou no seu tablet ou smartphone;

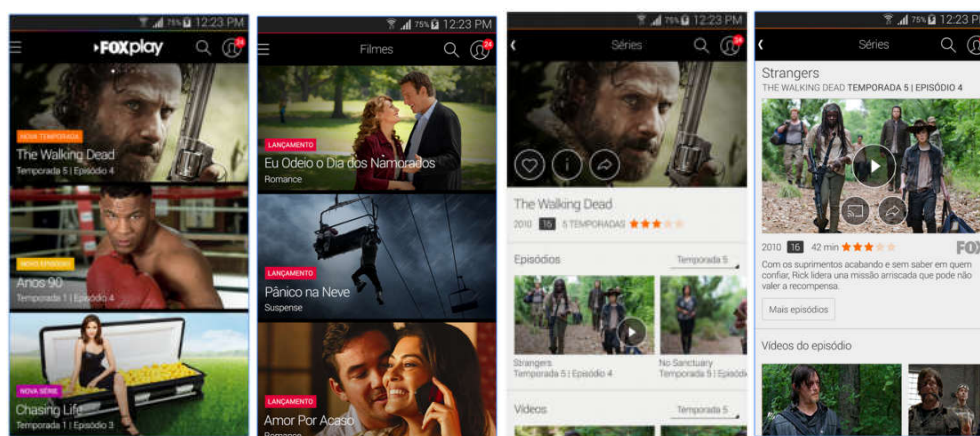
²³ <http://www.foxplaybrasil.com.br/>

²⁴ <https://play.google.com/store/apps/details?id=com.moviecity.app>

- Criar alerta de notificação para não perder seus eventos esportivos favoritos;
- Receber recomendações personalizadas.

O aplicativo FOX Play trabalha com empresas de publicidade móvel ou entidades similares, que trazem a publicidade relevante aos interesses do telespectador. Essa inteligência publicitária se deve ao *Ad Choices Link*²⁵ empresa que traça os interesses dos usuários a partir de aplicativos nos dispositivos móveis.

Figura B.10 - Telas retiradas do aplicativo FOX Play



A emissora Televisa²⁶ é a principal emissora de televisão mexicana, tornou-se conhecida no mundo inteiro por exportar programas de televisão, em especial telenovelas, a empresa produz atualmente cerca de 10 telenovelas por ano. Televisa não é um canal de televisão, mas sim um conglomerado de quatro canais, dois em rede nacional, uma semi-rede e outro canal local. Possui alguns aplicativos para dispositivos móveis. No entanto, será destacada apenas o que tem características de Segunda Tela.

²⁵ Extraído de: www.aboutads.info/appchoices

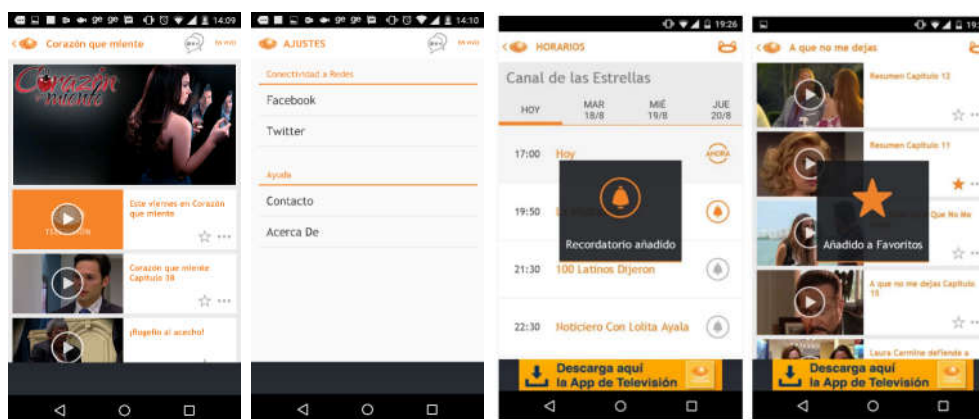
²⁶ <http://www.televisa.com/>

O app Television²⁷ oferece o melhor conteúdo de Programas e telenovelas especiais. Os capítulos são atualizados diariamente caso queira assistir novamente.

Com o app Televisa mostrado na Figura B.11 o telespectador poderá:

- Compartilhar os vídeos com os amigos em redes sociais;
- Conferir a programação e adicionar lembretes para programa de calendário;
- Marcar o conteúdo favorito;
- Assistir eventos totalmente ao vivo e o telespectador pode interagir com eles.

Figura B.11 - Telas retiradas do aplicativo Television



A emissora Entertainment and Sports Programming Network²⁸ (ESPN) é uma rede de TV por assinatura dos Estados Unidos dedicada à transmissão e produção de programas esportivos 24 horas por dia. Possui diversos aplicativos para dispositivos móveis, no entanto somente aplicativo para Segunda Tela serão apresentados a seguir.

O aplicativo ESPN Sync²⁹ traz sincronismo com a programação do canal, redes sociais, interatividade, gameificação, chat com seus amigos, estatísticas atualizadas em

²⁷ <https://play.google.com/store/apps/details?id=mx.naat.televisa.video>

²⁸ <http://espn.go.com/>

²⁹ <https://play.google.com/store/apps/details?id=com.mobovivo.espn>

tempo real, informações especiais dos jogadores, eventos internacionais vistos de um outro ângulo, vídeos e replay.

Com o app ESPN Sync mostrado na Figura B.12 o telespectador poderá:

- Acompanhar o placar dos principais jogos do futebol;
- Veja vídeos com gols e lances;
- Rever em vídeo jogadas e pontos marcantes da Associação Nacional de Basquetebol dos Estados Unidos (NBA) e Liga Nacional de Futebol Americano (NFL);
- Acompanhar times e esportes favoritos;
- Responder as enquetes, participe do aplicativo e marque pontos no seu ranking;

Figura B.12 - Telas retiradas do aplicativo ESPN Sync

