

TV 3.0: Integração e Controle de Renderizadores de Efeitos Sensoriais

Marina Ivanov

marinaivanov@midia.com.uff.br

Laboratório MídiaCom

Universidade Federal Fluminense

Niterói, Rio de Janeiro, Brasil

Joel A. F. dos Santos

jsantos@eic.cefet-rj.br

MultiSenS, Centro Federal de Educação Tecnológica

Celso Suckow da Fonseca

Rio de Janeiro, Rio de Janeiro, Brasil

Rômulo Vieira

romulo_vieira@midia.com.uff.br

Laboratório MídiaCom

Universidade Federal Fluminense

Niterói, Rio de Janeiro, Brasil

Débora C. Muchaluat-Saade

debora@midia.com.uff.br

Laboratório MídiaCom

Universidade Federal Fluminense

Niterói, Rio de Janeiro, Brasil

ABSTRACT

Advancements in the Brazilian television sector have been consistently intertwined with the technological innovations of each era. From its inception, supporting only live broadcasts, through the integration of recorded content on magnetic tapes, to the adoption of a color standard, television broadcasts have always been driven by the pursuit of higher quality. Currently, the Brazilian Digital Terrestrial Television System Forum (SBTVD) is working on a new national standard, referred to as TV 3.0, which, among various guidelines, aims to incorporate multiple secondary devices, such as sensory effect actuators, to enhance immersion and provide viewers with a richer content experience. In this context, the present paper proposes architectural changes to the TV receiver, a communication protocol, and an Application Programming Interface (API) to control various functions in sensory effect display devices, such as activation, deactivation, and adjustments in renderings. Additionally, a use case is presented to validate these concepts and demonstrate how they may find full applicability in the next generation of digital terrestrial television in Brazil.

KEYWORDS

Efeitos sensoriais, Middleware Ginga, NCL, HTML5, TV Imersiva

1 INTRODUÇÃO

A transmissão de conteúdo televisivo no Brasil passou por grandes transformações nos últimos anos. A transição da TV analógica para a digital disponibilizou novos recursos funcionais, como integração *broadcast-broadband* e a criação do chamado perfil D (ABNT 15606-1) [10], permitindo vídeo sob demanda e sincronização multitelas. Essas funcionalidades abriram caminho para novos serviços, permitindo que os espectadores comesçassem a demandar experiências personalizadas que atendessem às suas necessidades individuais. Em paralelo a isso, a ubiquidade de *smartphones*, computadores,

tablets e outros dispositivos digitais na vida contemporânea, levaram os produtores de televisão a reavaliarem suas abordagens de criação. Cada vez mais, os programas de TV são projetados para promover a interatividade, a transmidialidade, o uso de conteúdo 3D, *streaming* adaptativo e o consumo em segunda tela. Há também novas propostas para o aprimoramento da imersão por meio de realidade virtual [16]. Além disso, efeitos sensoriais, como aroma, vento, luz, etc., podem ser combinados com conteúdo multimídia tradicional para aperfeiçoar a imersão e a qualidade da experiência dos telespectadores [5, 7]. Em vista desse cenário, esses elementos tornaram-se requisitos do vanguardado padrão chamado de TV 3.0, especificado pelo Fórum SBTVD. A linguagem *Nested Context Language* (NCL) 4.0 [8, 9] foi escolhida como tecnologia para suporte a efeitos sensoriais na TV 3.0.

Entretanto, a recepção na TV digital pode se dar por diferentes equipamentos, como televisores ou *set-top boxes*. O mesmo ocorre para os atuadores dos efeitos sensoriais, que são dispositivos heterogêneos, desenvolvidos por diversos fabricantes e com as mais variadas funcionalidades, protocolos e tecnologias funcionais, o que dificulta uma comunicação padronizada com outros equipamentos presentes na mesma rede local. Somado a isso, em um ambiente de TV imersiva, os atuadores de efeitos podem ser controlados tanto por aplicações executadas diretamente na TV, quanto por aplicações das emissoras que são executadas em dispositivos de segunda tela.

A fim de viabilizar essas novas funcionalidades, especialmente o suporte a efeitos sensoriais, o presente trabalho propõe mudanças arquiteturais no receptor de TV 3.0, bem como um protocolo de comunicação entre o mesmo e dispositivos remotos que renderizam efeitos sensoriais e uma API para permitir o controle desses efeitos e sua sincronização com conteúdo audiovisual. Esta pesquisa segue a API genérica proposta em [16], que estabeleceu as diretrizes para comunicação entre dispositivos remotos heterogêneos e o receptor de TV 3.0, aqui aplicadas para inclusão e controle de renderizadores de efeitos sensoriais.

O restante do artigo está estruturado da seguinte maneira. A Seção 2 examina o estado da arte na integração entre conteúdo multimídia e efeitos sensoriais, abordando os métodos predominantes para combinar esses diferentes tipos de mídia e concentrando-se na aplicação desses parâmetros em ambientes de TV imersiva. A Seção 3 discute as modificações arquiteturais no receptor de TV

In: VII Workshop Futuro da TV Digital Interativa (WTVDI 2024). Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web (WTVDI'2024). Juiz de Fora/MG, Brazil. Porto Alegre: Brazilian Computer Society, 2024.
© 2024 SBC – Brazilian Computing Society.
ISSN 2596-1683

3.0 necessárias para suportar essas novas funcionalidades, além do protocolo de comunicação necessário para a descoberta e registro de dispositivos remotos, o que culmina na API capaz de controlar os efeitos sensoriais. A Seção 4 ilustra um caso de uso de conteúdo televisivo que incorpora efeitos de luz, aroma e vento, desenvolvido com base nos conceitos expostos neste estudo. Por fim, a Seção 5 apresenta as conclusões gerais decorrentes da realização deste trabalho.

2 TRABALHOS RELACIONADOS

A apresentação de conteúdo audiovisual acompanhada de efeitos que estimulam múltiplos sentidos humanos é conhecida como mulsemídia (do inglês, *Multiple Sensorial Media*) [2]. Essa abordagem permite a integração de efeitos sensoriais, tais como vento, aroma e temperatura, sincronizados com mídias tradicionais, como texto, vídeo, imagem e áudio. A renderização desses efeitos pode ser realizada por meio de dispositivos atuadores, como ventiladores, dispersores de aroma, lâmpadas e aquecedores [14].

Dada a ubiquidade das aplicações multimídia no cotidiano, diversos estudos foram conduzidos para integrar conteúdo de mídia tradicional com efeitos sensoriais em uma aplicação web. Um tipo recorrente de trabalho nesta área baseia-se no uso de *plug-ins*. Por exemplo, a proposta de Rainer et al. [12] apresenta uma extensão para navegadores web e uma biblioteca denominada Ambient Library (AmbientLib), que são responsáveis por extrair quadros de vídeo e informações temporais para sincronizar o conteúdo multimídia com efeitos sensoriais relacionados ao conteúdo do vídeo em questão. De forma similar, o trabalho de Waltl et al. [18] facilita a comunicação entre um navegador web executando conteúdo multimídia e o sistema amBX¹, formado por dois ventiladores, uma pulseira de vibração, duas luzes frontais e aparelhagem sonora. Este sistema também recupera quadros de vídeo e exibe uma cor correspondente no ambiente amBX, além de solicitar descrições de metadados dos efeitos sensoriais para ativar os ventiladores e a pulseira conforme o conteúdo exibido.

Além disso, há uma série de estudos focados em desenvolver *frameworks* para capturar, medir, quantificar, julgar e explicar a experiência do usuário no consumo de conteúdo mulsemídia. O trabalho de Waltl et al. [19] contribui ao realizar testes utilizando o padrão ITU-T P.911 e a técnica de (*Degradation Category Rating-DCR*) para avaliar e categorizar o estado de deteriorização dos renderizadores de efeitos sensoriais. Saleme et al. [15] propõem uma estrutura mulsemídia interoperável que acomoda diferentes perfis de comunicação e conectividade, além dos mais distintos padrões de metadados de efeitos sensoriais conforme as necessidades dos aplicativos e dispositivos disponíveis no ambiente do usuário. Guedes et al. [4], por sua vez, apresentam uma estrutura de programação de alto nível que suporta interfaces de usuário multimodais em aplicações multimídia interativas, integrando elementos de entrada e saída, como gestos, reconhecimento de voz, conteúdo audiovisual, sintetizadores de fala e atuadores para implementar diferentes efeitos sensoriais.

Por outro lado, a proposta de Josué et al. [8] foca na modelagem abstrata de efeitos sensoriais em aplicações multimídia, utilizando a linguagem NCL (ITU-T H.761/ABNT NBR 15606) [11, 13] para

tratar os efeitos sensoriais como entidades de primeira classe, permitindo sua definição independente da instalação física que executará a aplicação. Dessa forma, os efeitos sensoriais são modelados como entidades multimídia, análogas à representação abstrata de conteúdo multimídia tradicional.

Especificamente no contexto de sistemas de TV, as aplicações de Jalal et al. [5–7] utilizam a pervasividade da Internet das Coisas (IoT) para habilitar um cenário de casa inteligente com dispositivos reais e consolidados no mercado. Para viabilizar esse cenário, a arquitetura do ambiente é dividida em quatro camadas, a saber: camada física, implementada na nuvem e composta por objetos capazes de acessarem a Internet; camada de virtualização, que desacopla a parte de *hardware* da representação de *software* baseada em nuvem a partir da criação de uma contraparte digital de qualquer entidade no mundo real; camada de agregação, responsável pela junção dos dados vindos de diferentes fontes para garantir um alto nível de reutilização; e camada de aplicação, onde os aplicativos do usuário são responsáveis pelo processamento e apresentação do conteúdo final.

3 AMPLIANDO AS FUNCIONALIDADES DO GINGA PARA RENDERIZAÇÃO DE EFEITOS SENSORIAIS

Esta seção oferece uma visão geral sobre a inserção de efeitos sensoriais no GINGA, abrangendo desde seu comportamento no documento de autoria da aplicação até as modificações arquiteturais e o protocolo de comunicação necessário para suportar dispositivos atuadores heterogêneos. Este panorama culmina na API que controla um conjunto de funções nesses dispositivos.

3.1 Inserindo Efeitos Sensoriais no GINGA

O *middleware* GINGA [17] foi desenvolvido para permitir que serviços de TV operem em diversos modelos de *set-top boxes*, apresentando uma arquitetura composta por subsistemas que desempenham funções específicas voltadas para a exibição de aplicações declarativas. Tais aplicações são criadas utilizando a linguagem NCL, adotada como padrão na autoria de aplicação para o sistema de TV brasileiro, capaz de descrever o comportamento espacial e temporal entre seus objetos, habilitar interação dos usuários e descrever o *layout* da apresentação em múltiplos dispositivos. Para prover suporte a aplicações mulsemídia, a linguagem foi recentemente estendida para a versão 4.0, de modo a permitir interação multimodal do usuário por meio de comandos de voz, gestos ou movimentos oculares [1], além de representar estímulos sensoriais (como luzes RGB/estroboscópicas, estímulos táteis, olfativos e etc.) como entidades de primeira ordem [8, 9], independentemente dos dispositivos físicos utilizados para sua renderização.

Dessa forma, o autor da aplicação pode usar os mesmos conceitos já utilizados para manipular nós de mídia para os nós de efeitos sensoriais. O elemento *<effect>* possui um conjunto de atributos para identificar e caracterizar a ocorrência do efeito sensorial, como o *id*, que identifica univocamente cada elemento sensorial dentro do documento NCL, o *type*, que especifica o tipo de efeito (podendo assumir os valores de efeito de luz, *flash*, temperatura, vento, vibração, *spray*, aroma e névoa), e o *descriptor*, que dispõe de

¹<http://www.ambx.com>

propriedades adicionais, como uma especificação de início ou fim da apresentação, por exemplo.

No que tange à localização de tais efeitos, ela pode ser especificada de duas maneiras. Primeiro, o autor da aplicação pode usar um sistema de coordenadas esféricas, no qual a região dos efeitos começa no ponto indicado pelos atributos azimutal e polar, enquanto que no segundo método são utilizados os atributos de largura e altura para indicar o tamanho da área a ser usada para renderizar o efeito.

Por fim, cabe destacar que a unidade de medida para o efeito luminoso é definida em lux, enquanto a intensidade de um efeito térmico é expressa em graus Celsius. No que concerne ao efeito do vento, a mensuração é feita de acordo com a escala Beaufort. Para a medição do efeito de vibração, utiliza-se a escala Hertz (Hz). A intensidade dos efeitos de spray, aroma e névoa pode ser descrita em termos de mililitros por hora (ml/h). Para evitar que o autor de uma aplicação NCL tenha que lidar com tantas unidades distintas, a intensidade de um efeito sensorial é especificada em uma escala relativa de 0 a 10.

3.2 Arquitetura do Receptor TV 3.0

Renderizadores de efeitos sensoriais podem ser desenvolvidos por diferentes fabricantes e implementar diferentes protocolos de controle. Deste modo, torna-se necessário a definição de uma interface de comunicação, a fim de permitir que estes diferentes dispositivos se comuniquem com o receptor de TV para a execução de aplicações multemídia.

A comunicação dos renderizadores de efeito com o *middleware* Ginga pode se dar através de uma API específica para cada renderizador, permitindo que o mesmo se comunique diretamente com o Ginga Common Core, conforme proposto por Josué et al. [9]. Neste cenário, os implementadores de *middleware* devem desenvolver suas soluções já acopladas aos fabricantes de renderizador de efeito.

A fim de possibilitar uma maior independência entre o *middleware* e os fabricantes de renderizadores, este trabalho propõe um novo componente, externo ao *middleware*, que se comunica com o Ginga através do Ginga CCWS, denominado *Sensory Effect (SE) Presentation Engine*, e especifica seu próprio protocolo de comunicação com os renderizadores de efeitos sensoriais, conforme apresentado na Figura 1.

O *middleware* Ginga deve manter um registro dos renderizadores de efeitos cadastrados, incluindo as configurações necessárias para controlá-los, seja via API do dispositivo ou através do Ginga CC WebServices. Isso pode ser realizado por meio de um arquivo de configuração que contenha informações como o endereço IP de um dispositivo conectado diretamente ao Ginga ou o identificador de um *SE Presentation Engine* que controla um conjunto de dispositivos.

3.3 Comunicação do SE Presentation Engine com o Ginga

Conforme citado anteriormente, a comunicação entre o *SE Presentation Engine* e o *middleware* Ginga é estabelecida através do Ginga CC Webservices. Como o *SE Presentation Engine* é um elemento externo ao receptor, ele deve utilizar o protocolo SSDP para determinar a existência de um terminal receptor TV 3.0 com suporte ao Ginga CC WebServices e obter o ponto de acesso às API fornecidas.

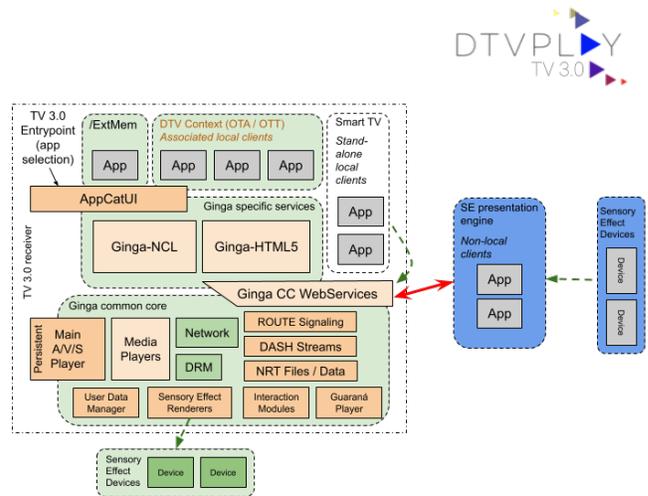


Figura 1: Arquitetura do *middleware* Ginga para suporte a controle de renderizadores de efeitos via Ginga CCWS.

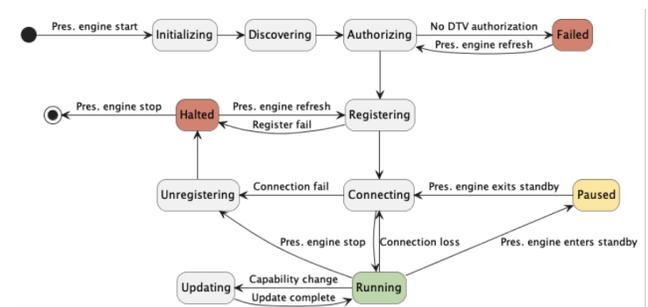


Figura 2: Processo de descoberta e registro de dispositivo remoto capaz de renderizar efeitos sensoriais [16].

O *Simple Service Discovery Protocol* (SSDP) [3] fornece um mecanismo para dispositivos em rede se comunicarem e se descobrirem. Para isso, o protocolo SSDP fornece suporte à descoberta *multicast*, bem como notificação baseada em servidor e roteamento de descoberta. Este processo de descoberta e registro de dispositivos remotos junto ao Ginga é descrito por Santos et al. [16] e ilustrado na Figura 2.

Após a descoberta, a comunicação avança para a fase de Autorização, onde o *Remote Device WebService* (RDWS), componente responsável por intermediar a comunicação entre o Ginga e os dispositivos remotos, obtém o *token* de acesso do dispositivo que está tentando se conectar e estabelece um vínculo com o receptor do sinal de TV. Em vista da necessidade de autorização do usuário, a etapa de Falha pode ser alcançada caso nenhuma autorização seja concedida.

Por outro lado, se a operação for autorizada, a fase de Registro é iniciada com o RDWS fazendo uma solicitação POST à API de registro para, em seguida, conectar-se ao ponto de entrada WebSocket, iniciando a fase de Conexão. Se algum erro for detectado,

a comunicação move-se para a fase de Cancelamento de Registro. Caso contrário, avança para a fase de Execução.

Para o registro do *SE Presentation Engine* no Ginga, o corpo da mensagem deve conter um campo que identifica a classe do dispositivo que está se registrando como renderizador de efeitos sensoriais e os tipos de efeitos suportados por este equipamento, que devem estar em conformidade com os efeitos suportados pelo *middleware*. A Listagem 1 apresenta um exemplo de mensagem para registro da *SE Presentation Engine*.

```

1 {
2   "deviceClass" : "sensory-effect",
3   "supportedTypes" : [ "LightType", "ScentType", "
4     WindType", "TemperatureType", "VibrationType" ]
5 }
```

Listagem 1: Corpo da mensagem de registro da *SE Presentation Engine*.

Na fase de Cancelamento de Registro, o RDWS cancela a operação no CCWS, liberando o WebSocket previamente criado. Este procedimento também pode ser acionado quando o mecanismo de apresentação do dispositivo remoto é interrompido. É possível mover-se desta fase para a etapa de Pausa, onde o registro pode ser reiniciado quando o dispositivo remoto é acionado pelo usuário, ou completamente encerrado, caso o mecanismo de apresentação do dispositivo remoto pare.

Na fase de Execução, o *middleware* pode enviar comandos para controlar a apresentação de efeitos sensoriais. A partir deste ponto, é possível retornar à fase de Conexão se a comunicação for perdida ou alcançar a fase de Pausa se o dispositivo entrar em modo de espera. O RDWS deve permanecer nesta fase durante toda a apresentação do nó no dispositivo remoto. Além disso, durante este período, o RDWS deve trocar mensagens com o Ginga (via WebSocket) para que os metadados auxiliem no controle dessa apresentação e notifiquem transições de eventos. Se ocorrerem alterações nas capacidades do dispositivo remoto durante esse processo, o RDWS deve transitar para a fase de Atualização.

A *SE Presentation Engine* pode ser invocada pelo Ginga para a reprodução de efeitos sensoriais de uma aplicação NCL 4. Neste caso, ao iniciar a aplicação NCL4, o Ginga deve enviar uma mensagem conforme apresentada na Listagem 2, contendo o identificador do nó de efeito sensorial na aplicação NCL, suas propriedades e também o identificador da aplicação.

```

1 {
2   "nodeId" : <nodeId>,
3   "type" : "<LightType | ScentType | WindType |
4     TemperatureType | VibrationType >"
5   "appId" : <appId>,
6   "properties" : [
7     { "name" : <propName>, "value" : <propValue> }
8   ]
9 }
```

Listagem 2: Mensagem com metadados de efeito sensorial a ser renderizado pela *SE Presentation Engine*.

Quando a *SE Presentation Engine* recebe uma solicitação do Ginga para realizar uma ação sobre determinado efeito sensorial, ela deve enviar uma mensagem de resposta indicando se a ação

foi bem-sucedida (retornando o código 200), ou se ocorreu alguma falha, seja pelo fato do tipo de efeito sensorial não ser suportado ou devido a alguma limitação do *SE Presentation Engine*. Caso a renderização do efeito sensorial seja interrompida por alguma interação externa ao Ginga, a *SE Presentation Engine* deve notificar o *middleware* sobre uma ação de *abort* no evento de apresentação do efeito em questão.

Como os dispositivos de renderização podem possuir capacidades e características diferentes, este trabalho também propõe uma API para que o Ginga possa solicitar tais informações à *SE Presentation Engine*, conforme descrito na Listagem 3

```

1 {
2   "type" : <effectType>,
3   "capabilities" : [ { "name" : <capName > } ]
4 }
```

Listagem 3: Mensagem de solicitação de capacidades da *SE Presentation Engine* realizada pelo Ginga.

3.4 API CCWS para Suporte a Efeitos Sensoriais via Ginga-HTML5

O *middleware* Ginga oferece suporte à execução de aplicações especificadas tanto na linguagem NCL quanto em HTML5. Embora NCL4 contemple, de forma nativa, a especificação de efeitos sensoriais, o HTML5 permite a especificação apenas de nós de mídia tradicionais, como vídeo e áudio. Assim, com o intuito de viabilizar a especificação de aplicações multimídia também para esta linguagem, este trabalho propõe uma API, denominada *sensory-effect-renderers*, para o controle de renderizadores de efeitos sensoriais por meio do Ginga CCWS.

A aplicação HTML5 pode consultar os renderizadores de efeitos disponíveis no receptor por meio de uma requisição GET na rota `http(s)://<host>/dtv/sensory-effect-renderers`. Em caso de sucesso, o Ginga CCWS responde à requisição com informações sobre os renderizadores disponíveis, como localização, efeitos suportados e também seu estado (isto é, se estão em uso por alguma aplicação, preparados, ou disponíveis, por exemplo). Caso a aplicação queira obter a informação de um renderizador específico, ela deve utilizar a requisição GET na rota `http(s)://<host>/dtv/sensory-effect-renderers/<renderer-id>`, onde *renderer-id* define o identificador do renderizador desejado.

O controle de um renderizador pela aplicação HTML5 se dá pela requisição POST na rota `http(s)://<host>/dtv/sensory-effect-renderers/<renderer-id>` com o corpo da mensagem especificando o tipo de ação a ser aplicada sobre o efeito sensorial, conforme descrito na Listagem 4.

```

1 {
2   "effectType" : "<effectType >",
3   "action" : <prepare | start | pause | resume | stop | set
4     >,
5   "properties" : [
6     { "name" : <propName>, "value" : <propValue> }
7   ]
8 }
```

Listagem 4: Mensagem de controle de um renderizador de efeito específico.

4 CASO DE USO

Esta seção apresenta um novo caso de uso viabilizado pelas APIs propostas neste trabalho: uma aplicação multimídia em HTML5 que abrange os requisitos da TV 3.0, especificamente relacionados ao suporte para TV imersiva (AP-req-14.1, 14.4 e 14.6).

Considere uma aplicação mulsemídia composta de três conteúdos de vídeo, efeito de luz e efeito de aroma. Inicialmente, é apresentado um efeito de luz laranja, refletindo a cor predominante no conteúdo audiovisual sendo exibido na tela. Durante a execução da aplicação, o efeito de luz deve ter a sua propriedade "color" alterada, sincronizada com a troca entre os conteúdos de vídeo que são apresentados na tela. Além do efeito de luz, a aplicação apresenta um efeito de aroma sincronizado com o terceiro vídeo. O efeito de aroma é renderizado pelo dispositivo no momento em que este vídeo começa a ser exibido. A Figura 3 exibe a visão temporal da aplicação, ilustrando a sincronização entre os vídeos e os efeitos de luz e aroma.

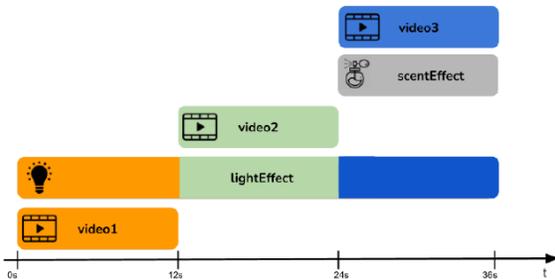


Figura 3: Linha do tempo correspondente ao plano de apresentação da aplicação de exemplo.

Para este caso de uso, será utilizada a rota de controle de renderizador de efeito apresentado na Seção 3.4 para disparar os efeitos sensoriais sincronizados com a apresentação dos elementos de vídeo. A Listagem 5 descreve a aplicação proposta.

```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4   ...
5 </head>
6 <body>
7   <video id="myVideo" controls>
8     <source src="media/video1.mp4" type="video
9     /mp4">
10  </video>
11  <script>
12    const videos = [
13      'media/video1.mp4',
14      'media/video2.mp4',
15      'media/video3.mp4'
16    ];
17
18    let currentIndex = 0;
19    const videoElement = document.
20    getElementById('myVideo');
```

```

21 videoElement.addEventListener('ended',
22 function() {
23   currentIndex++;
24   if (currentIndex < videos.length)
25   {
26     videoElement.src = videos[
27     currentIndex];
28     videoElement.play();
29   }
30 });
31
32 videoElement.addEventListener('play',
33 function() {
34   switch (currentIndex){
35     case 0:
36       startLightEffect("orange");
37       break;
38     case 1:
39       setLightColor("green");
40       break;
41     case 2:
42       setLightColor("blue");
43       startScentEffect("sea");
44       break;
45   }
46 });
47
48 videoElement.src = videos[
49 currentIndex];
50 videoElement.play();
51 </script>
52 </body>
53 </html>
```

Listagem 5: Aplicação HTML5 mulsemídia.

A função `startLightEffect` recebe a cor do efeito de luz como parâmetro, e implementa a requisição do tipo POST na rota `https://<host>/dtv/sensory-effect-renderers/<renderer-id>` com a mensagem de corpo descrita na Listagem 6.

```

1 {
2   "effectType": "LightType",
3   "action": "start",
4   "properties": [
5     {"name": "color", "value": "orange"}
6   ]
7 }
```

Listagem 6: Mensagem enviada pela função `startLightEffect`.

Por fim, as funções `setLightColor` e `startScentEffect` também utilizam a mesma rota da função `startLightEffect` porém com o corpo da mensagem contendo informações diferentes, conforme descrito nas Listagens 7 e 8, respectivamente.

```

1 {
2   "effectType": "LightType",
3   "action": "set",
4   "properties": [
5     {"name": "color", "value": "green"}
6   ]
7 }
```

Listagem 7: Mensagem enviada pela função setLightColor.

```

1 {
2   "effectType": "ScentType",
3   "action": "start",
4   "properties": [
5     { "name": "scent", "value": "sea" }
6   ]
7 }

```

Listagem 8: Mensagem enviada pela função startScentEffect.**5 CONSIDERAÇÕES FINAIS**

A evolução constante do setor televisivo brasileiro, impulsionada por inovações tecnológicas, continua a moldar a forma como o conteúdo é recebido pelos telespectadores. Este artigo apresentou uma proposta de adaptação do *middleware* Ginga para suportar a próxima geração de TV digital no Brasil, conhecida como TV 3.0, com foco na integração de renderizadores de efeitos sensoriais. Através de mudanças arquiteturais e a implementação de uma API específica, demonstramos como o Ginga pode se comunicar e controlar dispositivos remotos responsáveis pela renderização de efeitos como luz, aroma e vento, proporcionando uma experiência mais imersiva e interativa.

Os resultados obtidos a partir do caso de uso validaram a aplicabilidade das modificações propostas, evidenciando a viabilidade técnica e os benefícios proporcionados por uma experiência de TV sensorial, que além de aprimorar a imersão, abre novas possibilidades para o desenvolvimento de aplicações inovadoras no contexto da TV digital, como conteúdos publicitários mais impactantes e ambientes domésticos transformados em experiências 4D.

No entanto, a implementação completa e a adoção massiva desse novo padrão dependem de um ecossistema robusto que inclua fabricantes de dispositivos, emissoras e desenvolvedores de conteúdo. Futuras pesquisas irão focar em otimizações de desempenho, segurança da comunicação entre dispositivos e a criação de ferramentas que facilitem o desenvolvimento de aplicações sensoriais.

A TV 3.0 representa um salto significativo na experiência televisiva, trazendo novas formas de interatividade e imersão. Outrossim, este trabalho contribui para esse avanço, estabelecendo as bases tecnológicas para a integração de efeitos sensoriais no ecossistema de TV digital do Brasil.

AGRADECIMENTOS

Os autores agradecem o suporte das agências CAPES, Capes-Print, RNP, CNPq e FAPERJ, bem como ao Ministério das Comunicações (MCOM) e ao Fórum SBTVD.

REFERÊNCIAS

- [1] Fábio Barreto, Raphael Abreu, Marina Josué, Eyre Montevecchi, Pedro Valentim, and Débora Muchaluat-Saade. 2023. Providing multimodal and multi-user interactions for digital tv applications. *Multimedia Tools and Applications* 82 (2023), 4821–4846. <https://doi.org/10.1007/s11042-021-11847-3>
- [2] Gheorghita Ghinea, Christian Timmerer, Weisi Lin, and Stephen R Gulliver. 2014. Mulsemedia: State of the art, perspectives, and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11, 1s (2014), 1–23.
- [3] Y. Y. Goland, T. Cai, P. Leach, and Y. Gu. 1999. Simple service discovery protocol/1.0 operating without on arbiter. *IETF INTERNET-DRAFT draft-cai-ssdp-v1-03.txt* (1999). <https://datatracker.ietf.org/doc/html/draft-cai-ssdp-v1-03>
- [4] Alan Livio Vasconcelos Guedes, Roberto Gerson de Albuquerque Azevedo, and Simone Diniz Junqueira Barbosa. 2017. Extending multimedia languages to support multimodal user interactions. *Multimedia tools and applications* 76 (2017), 5691–5720.
- [5] Lana Jalal, Matteo Anedda, Vlad Popescu, and Maurizio Murrioni. 2018. Internet of Things for enabling multi sensorial TV in smart home. In *2018 IEEE Broadcast Symposium (BTS)*. IEEE, 1–5.
- [6] Lana Jalal, Matteo Anedda, Vlad Popescu, and Maurizio Murrioni. 2018. Qoe assessment for broadcasting multi sensorial media in smart home scenario. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 1–5.
- [7] Lana Jalal, Matteo Anedda, Vlad Popescu, and Maurizio Murrioni. 2018. QoE assessment for IoT-based multi sensorial media broadcasting. *IEEE Transactions on Broadcasting* 64, 2 (2018), 552–560.
- [8] Marina Josué, Raphael Abreu, Fábio Barreto, Douglas Mattos, Glauco Amorim, Joel dos Santos, and Débora Muchaluat-Saade. 2018. Modeling sensory effects as first-class entities in multimedia applications. In *Proceedings of the 9th ACM Multimedia Systems Conference*. 225–236.
- [9] Marina Josué, Marcelo F. Moreno, and Débora Muchaluat-Saade. 2024. Automatic Preparation of Sensory Effects. In *Proceedings of MMSys '24: ACM Multimedia Systems Conference 2024* (Bari), 35–40.
- [10] ABNT NBR. 2023. ABNT NBR 15606-1, Codificação de dados e especificações de transmissão para radiodifusão digital, Parte 1: Codificação de dados. *ABNT* (2023).
- [11] ABNT NBR. 2023. ABNT NBR 15606-2. Televisão digital terrestre—Codificação de dados e especificações de transmissão para radiodifusão digital—Parte 2: Ginga-NCL para receptores fixos e móveis—Linguagem de aplicação XML para codificação de aplicações. *ABNT* (2023).
- [12] Benjamin Rainer, Markus Waltl, Eva Cheng, Muawiyath Shujau, Christian Timmerer, Stephen Davis, Ian Burnett, Christian Ritz, and Hermann Hellwagner. 2012. Investigating the impact of sensory effects on the quality of experience and emotional response in web videos. In *2012 Fourth International Workshop on Quality of multimedia experience*. IEEE, 278–283.
- [13] ITU-T Recommendation. 2014. Nested Context Language (NCL) and Ginga-NCL. (2014). <https://www.itu.int/itu-t/recommendations/rec.aspx?rec=H.761>
- [14] Renato O Rodrigues, Marina IP Josué, Raphael S Abreu, Glauco F Amorim, Débora C Muchaluat-Saade, and Joel AF dos Santos. 2019. A proposal for supporting sensory effect rendering in ginga-ncl. In *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web*. 273–280.
- [15] Estêvão Bissoli Saleme, Celso AS Santos, and Gheorghita Ghinea. 2019. A mulsemia framework for delivering sensory effects to heterogeneous systems. *Multimedia Systems* 25 (2019), 421–447.
- [16] Joel Santos, Rômulo Vieira, , Marina Josué, Karen Oliveira, and Débora C. Muchaluat-Saade. 2024. Multidevice support in the Next Generation of the Brazilian Terrestrial TV System. In *Proceedings of the 2024 ACM International Conference on Interactive Media Experiences (Stockholm, Sweden) (IMX '24)*. Association for Computing Machinery, New York, NY, USA, 1–9. <https://doi.org/10.1145/3639701.3656304>
- [17] Luiz Fernando Gomes Soares, Marcio Ferreira Moreno, Carlos de Salles Soares Neto, and Marcelo Ferreira Moreno. 2010. Ginga-NCL: Declarative middleware for multimedia IPTV services. *IEEE Communications Magazine* 48, 6 (2010), 74–81.
- [18] Markus Waltl, Benjamin Rainer, Christian Timmerer, and Hermann Hellwagner. 2011. Sensory Experience for Videos on the Web. In *2011 Workshop on Multimedia on the Web*. 49–51. <https://doi.org/10.1109/MMWeb.2011.12>
- [19] Markus Waltl, Christian Timmerer, and Hermann Hellwagner. 2010. Increasing the user experience of multimedia presentations with sensory effects. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. IEEE, 1–4.