

objetivo principal é aumentar a Qualidade de Experiência (QoE) do usuário e proporcionar um conteúdo verdadeiramente imersivo [5]. Aplicações mulsemídia já são encontradas comercialmente em cinemas 4D, jogos digitais imersivos e simuladores.

A importância dos efeitos sensoriais é evidente ao ver que as normas da TV 3.0 incorporam o suporte à TV imersiva como um de seus requisitos primários [2]. Isso significa que as futuras aplicações de TV poderão especificar e sincronizar efeitos como vento, aroma, luz, vibração e temperatura com o conteúdo audiovisual exibido. Por exemplo, durante a transmissão de uma cena de praia, a aplicação pode comandar a renderização de um efeito de aroma marinho e de vento, emulando a brisa do mar para o telespectador. Essa capacidade de engajar múltiplos sentidos visa transformar a experiência televisiva, tornando-a mais envolvente e informativa [3].

O middleware Ginga, em sua especificação para a TV 3.0, suporta um ecossistema dual para o desenvolvimento de aplicações, baseado tanto em NCL quanto em HTML5. Pesquisas e trabalhos sobre o padrão têm avançado significativamente na especificação de efeitos sensoriais para o ambiente declarativo NCL [4], onde a versão 4.0 da linguagem já modela estímulos sensoriais, como os de luz, aroma e vento, como entidades de primeira ordem, com atributos e propriedades bem definidas. Essa evolução, contudo, criou uma disparidade funcional, visto que o ambiente HTML5 permaneceu limitado à renderização de mídias convencionais.

Para endereçar essa disparidade funcional, este trabalho foca no projeto e validação da SEPE, componente proposto em [4] e previsto na norma da TV 3.0 para viabilizar a comunicação com atuadores de efeitos. A arquitetura proposta posiciona a SEPE como uma camada de abstração que, por meio de uma API, permite às aplicações HTML5 orquestrar experiências mulsemídia. Dessa forma, esta pesquisa oferece uma solução concreta buscando, assim, equiparar as capacidades imersivas entre os dois tipos de aplicação suportados pelo Ginga.

3 ARQUITETURA PROPOSTA

O principal objetivo da SEPE é abstrair a complexidade de comunicação com diferentes dispositivos de efeitos sensoriais, oferecendo uma interface única para o middleware Ginga. Para alcançar a flexibilidade e a extensibilidade necessárias, a arquitetura da SEPE foi projetada de forma modular, com destaque para a aplicação do padrão de projeto *Adapter*. O padrão permite que a SEPE se comunique com qualquer dispositivo, independentemente de seu protocolo proprietário. A lógica consiste em desenvolver um adaptador de software específico para cada novo tipo de dispositivo. Esse adaptador é responsável por traduzir os comandos genéricos recebidos da SEPE para os comandos específicos que o hardware do dispositivo entende. Essa abordagem garante que a plataforma seja facilmente extensível, bastando criar um novo adaptador sem modificar o núcleo do sistema¹.

Para garantir a interoperabilidade e o funcionamento correto do sistema, cada adaptador deve obrigatoriamente seguir uma estrutura bem definida, estabelecida por meio de um "contrato formal". Este contrato é materializado pela interface *IEffectRenderer*, que

todo adaptador precisa implementar. A função dessa interface é padronizar a forma como o núcleo da SEPE se comunica com os adaptadores, garantindo que, independentemente do dispositivo final ou de sua tecnologia proprietária, a interação com o sistema central seja sempre consistente e previsível. Assim, o sistema se mantém desacoplado das implementações concretas de cada dispositivo.

Este contrato especifica os elementos essenciais que todo renderizador de efeito deve possuir. A interface *IEffectRenderer* estabelece que cada adaptador deve, necessariamente, conter uma lista de tipos de efeitos suportados, uma descrição de suas capacidades (como estado de execução e localização relativa ao espectador) e, fundamentalmente, um método *handleCommand* para receber e processar os comandos de acionamento vindos da SEPE. Embora o desenvolvedor tenha total liberdade na implementação de como ele se comunica com o hardware, sua estrutura externa deve aderir rigorosamente a essa interface para garantir a integração bem-sucedida ao ecossistema.

O desenvolvimento e os testes deste trabalho foram realizados sobre o protótipo da *Application-oriented Platform* (AoP), uma plataforma que simula o comportamento do middleware Ginga e dos serviços da TV 3.0 em um ambiente de desktop controlado².

3.1 Fluxo de Comunicação

O processo de acionamento de um efeito sensorial segue um fluxo de comunicação bem definido que atravessa diferentes componentes do ecossistema TV 3.0. As etapas deste fluxo são as seguintes:

- (1) **Requisição pela Aplicação:** A aplicação de TV (App HTML), em execução no Ginga, envia uma requisição de comando (via HTTP POST) para o módulo TV 3.0 WebServices, especificando qual efeito deseja acionar, a ação e suas propriedades (ex: ligar luz, cor azul).
- (2) **Encaminhamento pelo Ginga:** O TV 3.0 WebServices recebe a requisição e direciona o comando para o canal de comunicação WebSocket que está conectado de forma persistente com a SEPE.
- (3) **Processamento na SEPE:** A SEPE, ao receber a mensagem via WebSocket, utiliza seu controlador interno (Device Controller) para processar o comando. O controlador identifica qual adaptador é o correto para o tipo de efeito solicitado (ex: adaptador de luz para um *LightType*).
- (4) **Tradução pelo Adaptador:** O comando genérico é repassado ao adaptador correspondente. Neste ponto, o adaptador traduz o comando para o protocolo específico do dispositivo físico que ele controla.
- (5) **Acionamento do Dispositivo Remoto:** Por fim, o adaptador envia o comando já traduzido para o dispositivo final (a lâmpada, o difusor de aroma, etc.), que executa a ação requisitada.

A Figura 2 demonstra o fluxo de comunicação do comando, da aplicação ao dispositivo renderizador do efeito sensorial, no caso uma lâmpada inteligente. Uma vez que a SEPE está conectada, o comando solicitando a ativação da luz na cor azul é enviada para o TV 3.0 WebServices que, por sua vez, repassa a mensagem pelo canal WebSocket para a SEPE. A SEPE se encarrega de identificar

¹O código-fonte pode ser acessado em: <https://github.com/motadv/se-presentation-engine>

²O código-fonte pode ser acessado em: <https://github.com/GPMM/GingaDistrib>

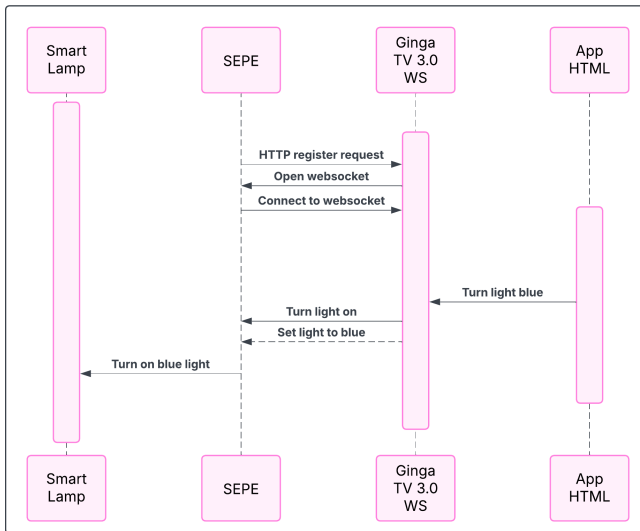


Figura 2: Representação gráfica do fluxo de comando para ativação de um efeito sensorial por uma aplicação GINGA-HTML5.

qual dispositivo inteligente é capaz de executar esse comando pelo seu tipo de registro (nesse exemplo, uma luz com o tipo suportado sendo "LightType") e envia o comando para o adaptador do dispositivo. Nele o comando é traduzido para o protocolo proprietário da lâmpada e o comando é, finalmente, executado.

4 VALIDAÇÃO EXPERIMENTAL

Para validar a arquitetura e demonstrar a funcionalidade do sistema, foi desenvolvido um estudo de caso prático simulando um ambiente doméstico conectado. O cenário de teste consistiu na implementação de dois componentes principais, sendo eles uma aplicação HTML e dois adaptadores para dispositivos de efeitos sensoriais.

Aplicação de Demonstração (TV UFF): Uma aplicação web simples, em HTML e Javascript, foi criada para ser executada no protótipo AoP, como representado na Figura 3. Essa aplicação realiza o descobrimento dos efeitos sensoriais disponíveis e envia comandos para ativá-los, consumindo a API de Web Services da TV 3.0³.

Adaptadores de Dispositivos Heterogêneos: Para comprovar a extensibilidade e a flexibilidade da SEPE, foram implementados dois adaptadores para dispositivos com modelos de comunicação completamente distintos:

- **Lâmpada Inteligente Xiaomi Yeelight:** Representa um dispositivo controlado em rede local. O adaptador se comunica diretamente com a lâmpada via socket TCP, enviando comandos em seu protocolo proprietário para ligar, desligar e alterar cor e intensidade.
- **Difusor de Aromas Moodo:** Representa um dispositivo controlado via nuvem. O adaptador se comunica com uma

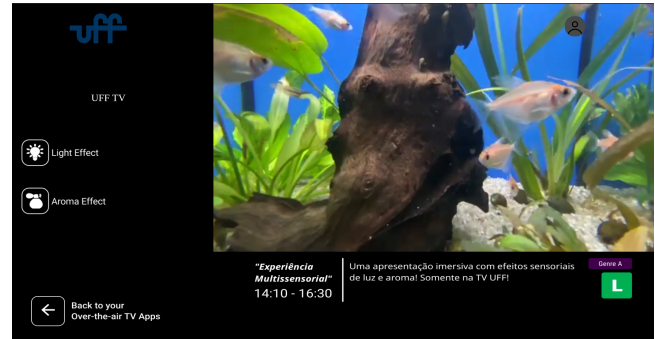


Figura 3: Tela da aplicação TV UFF para validação do sistema.

API RESTful externa do fabricante, necessitando de autenticação e conexão com a Internet para controlar a ativação das cápsulas de fragrância.

A implementação bem-sucedida de ambos os adaptadores e a capacidade da aplicação de TV de controlar os dois dispositivos através da SEPE validaram a eficácia da arquitetura proposta. O sistema demonstrou ser capaz de abstrair as complexidades de diferentes tecnologias de hardware e protocolos de comunicação, sejam eles locais ou baseados na nuvem.

5 AVALIAÇÃO DE DESEMPENHO

Foi realizada uma avaliação quantitativa com foco na latência de ponta a ponta do sistema, desde a solicitação na aplicação até a confirmação de execução pelo dispositivo físico. A medição foi realizada através da instrumentação do código, inserindo registros de tempo (*timestamps*) em pontos-chave da comunicação para comandos de controle.

As seguintes etapas foram cronometradas:

- **T0:** A aplicação HTML envia a requisição HTTP de controle.
- **T1:** O servidor Ginga TV 3.0 Web Services recebe a requisição em sua API.
- **T2:** O Ginga TV 3.0 Web Services envia a mensagem de comando via WebSocket para a SEPE.
- **T3:** A SEPE recebe a mensagem do WebSocket.
- **T4:** O adaptador da SEPE envia o comando final para o dispositivo físico.
- **T5:** O adaptador recebe a confirmação de sucesso do dispositivo.

A latência de cada segmento do fluxo foi calculada a partir da diferença entre os *timestamps* dessas etapas. Para cada cenário de teste, foram realizadas 30 medições para calcular a média e o desvio padrão. Os resultados para efeitos de luz e aroma estão apresentados nas Tabelas 1 e 2.

A análise dos resultados revela um desempenho altamente satisfatório. Para os efeitos de iluminação, a latência média de 8,966 ms é, na prática, imperceptível para o usuário final e abaixo de valores aceitáveis encontrados na literatura, onde foi medido um atraso de 0,9 segundos. [6]. Para os efeitos olfativos, a latência de 559,267 ms (cerca de meio segundo) [8] está confortavelmente dentro dos limites aceitáveis para esse tipo de estímulo, que segundo a literatura pode ter uma defasagem de até 15 segundos sem prejudicar a QoE.

³O vídeo da demonstração da aplicação TV UFF pode ser visualizado a partir do link: <https://www.youtube.com/watch?v=YGAELBzXfVo>

Tabela 1: Latência média para o cenário de efeito de luz.

| Etapa da Comunicação | Tempo(ms) | Desvio Padrão(ms) |
|-----------------------------|--------------|-------------------|
| Requisição da Aplicação | 6,655 | 5,970 |
| Processamento do Ginga | 0,172 | 0,448 |
| Trânsito no WebSocket | 1,379 | 1,278 |
| Processamento da SEPE | 0,103 | 0,315 |
| Resposta do Dispositivo | 0,655 | 0,573 |
| Latência Média Total | 8,966 | 6,489 |

Tabela 2: Latência média para o cenário de efeito de aroma.

| Etapa da Comunicação | Tempo(ms) | Desvio Padrão(ms) |
|-----------------------------|----------------|-------------------|
| Requisição da Aplicação | 5,167 | 4,981 |
| Processamento do Ginga | 0,167 | 0,384 |
| Trânsito no WebSocket | 0,767 | 0,841 |
| Processamento da SEPE | 0,167 | 0,677 |
| Resposta do Dispositivo | 552,8 | 49,797 |
| Latência Média Total | 559,267 | 51,018 |

É crucial notar que os principais gargalos de latência foram identificados em componentes externos à SEPE. No caso da lâmpada Yeelight, o maior atraso (6,655 ms) ocorreu no processamento interno do *framework* utilizada para executar o módulo da TV 3.0 WebServices, e no caso do Moodo, a maior parte do tempo (552,8 ms) foi consumida na comunicação com a API externa na nuvem. Isso comprova que o design da SEPE gera um overhead mínimo na comunicação, sendo um componente altamente eficiente e que o desempenho geral do ecossistema depende mais das tecnologias dos dispositivos e do middleware do que da própria SEPE.

6 CONCLUSÃO

Este trabalho apresentou um protótipo funcional e eficiente da *Sensory Effect Presentation Engine* (SEPE), um componente chave para habilitar experiências multissensoriais imersivas na TV 3.0. A arquitetura, baseada no padrão de projeto Adapter, provou ser robusta, flexível e extensível, integrando com sucesso dispositivos heterogêneos com modelos de comunicação distintos. A avaliação de desempenho demonstrou que a solução possui baixa latência, atendendo aos requisitos de QoE para a sincronização de efeitos sensoriais.

Como propostas de trabalhos futuros, destaca-se a implementação de um mecanismo de descoberta automática de dispositivos, para que a SEPE possa encontrar e registrar novos dispositivos na rede local de forma automática, sem configuração manual. Além disso, pretende-se realizar a integração com os módulos de privacidade e segurança da TV 3.0, para gerenciar permissões e garantir que apenas aplicações autorizadas possam controlar os dispositivos do usuário. Um outro trabalho interessante é a validação qualitativa com desenvolvedores, para avaliar a facilidade de criação de novos adaptadores, validando a extensibilidade da arquitetura na prática. Além disso, a calibração de renderizadores permitirá implementar um sistema que meça a latência de cada dispositivo e use

essa informação para calibrar o tempo de acionamento dos efeitos, garantindo uma sincronização ainda mais precisa com o conteúdo audiovisual.

REFERÊNCIAS

- [1] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. 202X. ABNT NBR 25608: TV 3.0 – Application coding. Projeto de norma em elaboração..
- [2] Fórum do Sistema Brasileiro de TV Digital Terrestre. 2020. *Call for Proposals: TV 3.0 Project*. Chamada de Propostas. Fórum do Sistema Brasileiro de TV Digital Terrestre (SBTVD). Disponível em: <https://forumsbtvd.org.br/wp-content/uploads/2020/07/SBTVDTV-3-0-CfP.pdf>. Acesso em: 9 jul. 2025..
- [3] Gheorghita Ghinea, Christian Timmerer, Weisi Lin, and Stephen R. Gulliver. 2014. Mulsemedia: State of the Art, Perspectives, and Challenges. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 1s, Article 17 (Oct. 2014), 23 pages. <https://doi.org/10.1145/2617994>
- [4] Marina Ivanov, Rômulo Vieira, Joel Santos, and Débora Muchaluat-Saade. 2024. TV 3.0: Integração e Controle de Renderizadores de Efeitos Sensoriais. In *Anais Estendidos do XXX Simpósio Brasileiro de Sistemas Multimídia e Web* (Juiz de Fora/MG). SBC, Porto Alegre, RS, Brasil, 297–302. https://doi.org/10.5753/webmedia_estendido.2024.244585
- [5] 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* (Amsterdam, Netherlands) (MMSys '18). Association for Computing Machinery, New York, NY, USA, 225–236. <https://doi.org/10.1145/3204949.3204967>
- [6] Marina Josué, Débora Muchaluat-Saade, and Marcelo Moreno. 2018. Preparation of Media Object Presentation and Sensory Effect Rendering in Mulsemedia Applications. In *Anais do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web* (Salvador). SBC, Porto Alegre, RS, Brasil, 45–52. <https://doi.org/10.1145/3243082.32430>
- [7] Ministério das Comunicações. 2024. *TV 3.0*. Retrieved July 9, 2025 from <https://www.gov.br/mcom/pt-br/assuntos/radio-e-tv-aberta/tv-30>
- [8] Niall Murray, Brian Lee, Yuansong Qiao, and Gabriel Miro-Muntean. 2017. The Impact of Scent Type on Olfaction-Enhanced Multimedia Quality of Experience. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 9 (2017), 2503–2515. <https://doi.org/10.1109/TSMC.2016.2531654>