

# Possibilitando o Reconhecimento de Expressões Faciais em Aplicações Ginga-NCL

Pedro Alves Valentim  
MídiaCom Lab - UFF  
Rio de Janeiro, Brazil  
pedroalvesvalentim@midia.com.uff.br

Fábio Barreto  
MídiaCom Lab - UFF  
UNILASALLE-RJ  
Rio de Janeiro, Brazil  
fbarreto@midia.com.uff.br

Débora C. Muchaluat-Saade  
MídiaCom Lab - UFF  
Rio de Janeiro, Brazil  
debora@midia.com.uff.br

## ABSTRACT

As the facial recognition research field grows, so do the possibilities for digital TV applications. However, in the current state of the art, it is not safe to assume there is a certain algorithm that would be the best for all kinds of applications. This work proposes an architecture to enable facial expression recognition for TV in a way that is agnostic to the recognition algorithm. As proof of concept, the proposal was developed for the Ginga *middleware*. There are two implementations: the first one, based on the current version of the Ginga *middleware*, and the second one, based on a proposed extended version of the *middleware*, exploring the viability of the present work.

## KEYWORDS

Facial Recognition, Affective Computing, Ginga, NCL, Multimedia Applications

## 1 INTRODUÇÃO

A implementação de técnicas para reconhecimento facial, imaginada em clássicos como *Blade Runner* e *Minority Report*, já é uma realidade. Tal recurso serve a um vasto leque de finalidades. Do ponto de vista de uma aplicação interativa, a expressão facial como dado de entrada pode ser mais veloz e mais fiel ao sentimento do usuário do que um clique de botão [11] [15] [16] [7] [31]. Isso pode ser particularmente interessante em aplicações muito dinâmicas, como as aplicações para TV, em que as reações do usuário podem variar tão subitamente quanto num susto. Além da identificação de expressões no rosto de um único usuário, é desejável distinguir também múltiplos usuários.

O Sistema Brasileiro de TV Digital (SBTVD) tem aberta uma proposta para a chamada TV 3.0<sup>1</sup>, a nova geração da TV aberta no Brasil. A chamada traz como um requisito a oferta de novas formas de interação com o usuário. Aplicações para o *middleware* Ginga, padrão do SBTVD [1] e padrão ITU para serviços IPTV [19], são escritas na linguagem NCL. NCL é uma linguagem declarativa para autoria de documentos hipermídia baseada em XML, com

<sup>1</sup>[https://forumsbtvd.org.br/tv3\\_0/](https://forumsbtvd.org.br/tv3_0/)

Os autores gostariam de agradecer a CNPq, CAPES e FAPERJ pelo apoio financeiro parcial neste trabalho.

In: XVII Workshop de Trabalhos de Iniciação Científica (WTIC 2020), São Luís, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2020.

© 2020 SBC – Sociedade Brasileira de Computação.  
ISSN 2596-1683

paradigma de sincronização baseado em eventos. Na versão atual, o *middleware* Ginga não dá suporte a reconhecimento facial.

Em acordo com o requisito da proposta para a TV 3.0 e contribuindo com pesquisas em aplicações multiusuário [4] [36], este trabalho propõe uma solução para reconhecimento de expressões faciais através do Ginga. Para tal, foram feitas duas implementações: a primeira usando a versão padrão do *middleware* e a segunda usando uma versão modificada, estendida, para que haja a especificação desse tipo de interação em documentos NCL.

Este artigo está estruturado como se segue. A Seção 2 trata de pesquisas nas áreas de computação afetiva e análise de sentimentos, reconhecimento facial e propostas de extensão ao *middleware* Ginga. A Seção 3 trata da arquitetura da solução proposta neste trabalho. A Seção 4 apresenta detalhes das implementações propostas. E, finalmente, a Seção 5 traz as considerações finais e propostas de trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

O reconhecimento facial é uma técnica que não exige dispositivos periféricos fora do cotidiano. A princípio, qualquer dispositivo de captura de vídeo (e. g. *webcam*, *smartphone*) é suficiente para captura do dado de entrada. Por isso, a relação custo-benefício costuma ser positiva, dada a vasta gama de aplicabilidades. Um caso típico é o uso em sistemas de vigilância e segurança em geral [27] [9], mas há inúmeros outros. Por exemplo, para foco automatizado e dinâmico em videoconferências [8], sistema de presença em sala de aula [20] e marcação assistida de pessoas em fotos [21].

Neste contexto, há uma aplicação que é o reconhecimento de expressões faciais. No entanto, os desafios são muito variados entre determinar o que configura um rosto [32], identificar um rosto [34] [23] e determinar, baseado em elementos de um rosto, a emoção que ele expressa [25] [28]. É importante notar que em cada etapa há menos consensos, mas mais espaço pra propostas. Isso é particularmente interessante, porque permite que pesquisas como a proposta no presente trabalho, em que o reconhecimento de expressões faciais não é o objeto principal da pesquisa, sejam flexíveis no que diz respeito à técnica aplicada.

Uma área que se beneficia diretamente do reconhecimento de expressões faciais é a computação afetiva. Um campo multidisciplinar e em expansão, a computação afetiva trata da compreensão e da emulação de emoções humanas em máquinas [29] [30] [31] [7]. Ter acesso à emoção do usuário como dado é, portanto, um de seus fundamentos. Falando especificamente em análise facial, também há vasta produção científica [10] [6] [26] [33].

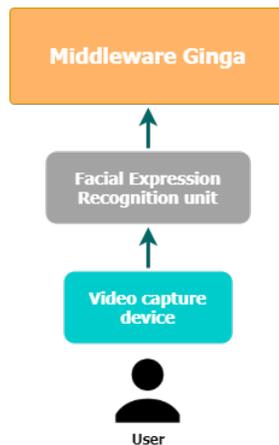
Com relação ao Ginga-NCL, há muitos trabalhos que podem dar e receber contribuições referentes à presente proposta. Alguns

exemplos, são trabalhos de extensão à NCL, como os de Barreto et al. [4] [3] e Guedes et al. [36], que apresentam propostas de funcionalidade multiusuário, que, por isso, se beneficiariam diretamente de reconhecimento facial. Outros exemplos são as proposições de extensão de Mello Brandão et al. [12] e Batista et al. [5], que tratam da comunicação com dispositivos de rede doméstica.

### 3 ARQUITETURA DA PROPOSTA

Sendo o conjunto de todas as possíveis expressões faciais difícil de delimitar [13] [14] [22], é natural que, em uma aplicação que faça reconhecimento de expressões faciais, o conjunto de expressões reconhecíveis seja previamente estabelecido. Em nossa proposta, o autor da aplicação define as expressões a serem reconhecidas.

Do mesmo modo, o autor é responsável por prover uma unidade de reconhecimento que seja, essencialmente, um programa que faça ponte entre o dispositivo de captura de vídeo, fazendo o reconhecimento das expressões faciais, e o *middleware*. A Figura 1 ilustra a arquitetura da solução proposta.



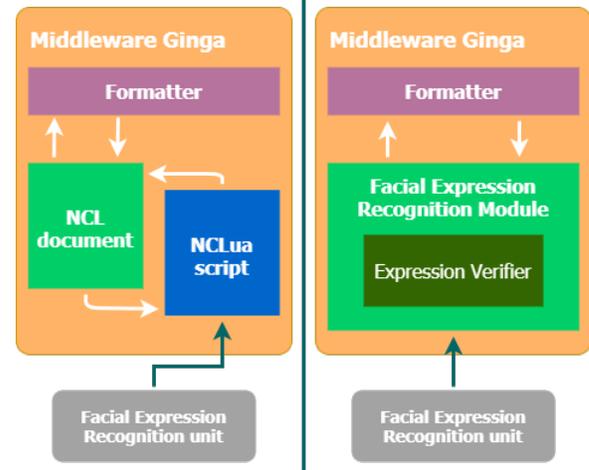
**Figura 1: Arquitetura da proposta, não-dependente de uma implementação específica da unidade de reconhecimento.**

A proposta deste trabalho, então, é a da adição do reconhecimento de expressões faciais em aplicações Ginga-NCL, usando uma unidade de reconhecimento arbitrária. Esse último ponto é importante porque, como discutido anteriormente, não há uma maneira correta, melhor, ou perfeitamente suficiente, de prover esse reconhecimento. Por isso, a proposta é agnóstica, permitindo que o autor possa usar a solução que deseje para o reconhecimento de expressões faciais.

O processo proposto é composto por poucas etapas. Durante a execução da aplicação, a face do usuário, capturada em vídeo, é constantemente processada pela unidade de reconhecimento. O resultado desse processamento, então, é direcionado ao *middleware* e, por fim, a aplicação NCL o recebe.

### 4 IMPLEMENTAÇÕES

São duas as implementações deste trabalho: uma, feita na versão atual do *middleware* Ginga-NCL, e outra, feita sobre uma proposta de extensão do *middleware* [2]. A Figura 2 ilustra a principal diferença entre as implementações, em termos de arquitetura.



**Figura 2: Arquiteturas das implementações. À esquerda, na versão atual do *middleware*; à direita, na versão estendida.**

Para ambas as implementações, a comunicação entre a unidade de reconhecimento e o *middleware* foi feita via MQTT [17]. MQTT foi escolhido por ser um protocolo baseado no paradigma *publisher/subscribe* extremamente leve.

Também para ambas as implementações, as unidades de reconhecimento facial usadas nos testes fazem o reconhecimento usando redes neurais convolucionais. Essa escolha foi feita com base em tendências da área [35] [24]. As unidades usadas nos testes eram capazes de identificar sete expressões faciais: neutra ("NEUTRAL"), irritada ("ANGRY"), aversa ("DISGUST"), assustada ("FEAR"), surpresa ("SURPRISE"), feliz ("HAPPY") e triste ("SAD").

#### 4.1 Implementação baseada no Ginga atual

A primeira implementação, fundamentada na versão atual do *middleware*, que não oferece nenhum tipo de interação por reconhecimento facial, foi feita usando eventos customizados em NCLua, isto é, em um script Lua [18] que roda em conjunto com a aplicação Ginga-NCL.

Para tal, é necessário um script NCLua dedicado que se conecte ao broker MQTT, como ilustrado na Listagem 1. O nó NCLua contém uma tabela de ações para cada expressão facial reconhecível, como mostrado na Listagem 2, e consulta as mensagens recebidas via MQTT periodicamente, conforme a Listagem 3, comparando-as com as entradas da tabela de ações, para potencialmente disparar outros eventos.

##### Listagem 1: Código NCLua para conexão via MQTT.

```
1 local client = mqtt.client.create(broker, port, callback)
2 client:connect('listener')
3 client:subscribe({'facial_expression_recognition'})
```

##### Listagem 2: Código NCLua que define o comportamento resultante do reconhecimento de cada expressão.

```

1 local actionTable = {
2   ['HAPPY'] = {
3     class = 'ncl',
4     type = 'presentation',
5     action = 'start',
6     label = 'videoAnchor',
7   },
8   ['SAD'] = {
9     class = 'ncl',
10    type = 'presentation',
11    action = 'stop',
12    label = 'videoAnchor',
13  },
14  -- (...)
15 }
16 local callback = function(topic, payload)
17   event.post(actionTable[payload])
18 end

```

Listagem 3: Código NCLua que invoca a si próprio a cada terço de segundo.

```

1 local handler = function(evt)
2   if (evt.class == 'ncl'
3     and evt.type == 'presentation'
4     and evt.action == 'start')
5     or (evt.class == 'user') then
6
7     local sec = 1000
8     event.timer(sec/3, function()
9       client:handler()
10      event.post('in', {class='user'})
11    end)
12  end
13 end
14 event.register(handler)

```

## 4.2 Implementação baseada no Ginga estendido

A segunda implementação é baseada em uma versão alternativa do *middleware*, que se beneficia de um novo tipo de evento especialmente projetado, chamado *FaceRecognition*. Durante o *parsing* do documento NCL, se houver um ou mais objetos interligados por elos, que usam conectores com papéis de condição iguais a *onFaceRecognition*, o formatador comunica ao *Facial Expression Recognition Module* quais expressões faciais são esperadas. Quando houver a detecção de tal expressão de um determinado usuário, o módulo informa ao formatador que o evento *FaceRecognition* ocorreu. No formatador, existe um processo que fica aguardando notificações dos módulos de interação.

O comportamento do evento *FaceRecognition* é similar ao do evento de seleção (*selection*) comandado pelo acionamento de teclas do controle remoto da TV. Considerando a máquina de estados de eventos da NCL, há uma transição instantânea do estado *prepared* para *occurring* e depois para *prepared*, quando uma expressão facial é reconhecida. Diferentemente de eventos de apresentação de mídia, em que o objeto pode ser iniciado, pausado ou finalizado durante a

execução, o evento *FaceRecognition* não pode ser manipulado em ações de elos NCL.

Dessa maneira, o papel exercido pelo script Lua na implementação anterior é feito pelo próprio *middleware*, exigindo apenas que o autor utilize uma condição que especifique a ocorrência do evento. Para isso, são necessários um conector que defina um papel de condição com atributo *role* igual a *onFaceRecognition*, onde o valor do atributo *key* indica o tipo de expressão facial a ser capturada, que pode ser parametrizado, como ilustrado na Listagem 4. Além disso, um elo, que utiliza esse conector, define exatamente o tipo de expressão facial desejada, conforme ilustrado na Listagem 5.

Listagem 4: Código de conector NCL que para a exibição de um componente ao ocorrer o evento *onFaceRecognition*

```

1 <causalConnector id="onFaceRecognitionStop">
2   <connectorParam name="key" />
3   <simpleCondition role="onFaceRecognition" key="$key" />
4   <simpleAction role="stop" />
5 </causalConnector>

```

Listagem 5: Código de elo NCL que para a mídia *video* ao reconhecer a expressão facial "SAD" utilizando o novo papel *onFaceRecognition*.

```

1 <link xconnector="onFaceRecognitionStop">
2   <bind role="onFaceRecognition" component="video">
3     <bindParam name="key" value="SAD"/>
4   </bind>
5   <bind role="stop" component="video"/>
6 </link>

```

## 4.3 Discussão

Contraopondo as implementações, com interesse em determinar se uma é favorável a outra, pode-se levar em conta dois fatores: a legibilidade e facilidade de autoria e o desempenho da implementação. Por autor entende-se alguém que tenha interesse em desenvolver uma aplicação Ginga-NCL.

Para autoria com a segunda implementação, só é necessário conhecimento de NCL. Por outro lado, a primeira implementação requer conhecimentos – não triviais – de Lua e NCLua. Por isso, toda aquela configuração mostrada na Seção 4.1 pode ser excessivamente complicada em comparação com a segunda.

No que diz respeito ao desempenho, este trabalho segue uma abordagem similar a [2], que apresenta uma proposta para interação em Ginga-NCL através de reconhecimento de voz. Em [2], os autores apresentam resultados mostrando que a segunda implementação integrada ao *middleware* Ginga tem desempenho superior à primeira implementação em NCLua. Como trabalhos futuros, serão realizados testes para comprovar a potencial superioridade de desempenho da segunda proposta.

## 5 CONSIDERAÇÕES FINAIS

O presente trabalho apresenta duas abordagens para fornecer o reconhecimento de expressões faciais em aplicações Ginga-NCL,

de modo que a técnica de reconhecimento seja totalmente customizável. Das abordagens, uma que pode ser usada na versão atual do *middleware* e outra usando – e endossando – uma versão estendida. A partir disso, espera-se incentivar o emprego da funcionalidade e fomentar pesquisa na área de computação afetiva, no contexto da televisão.

É importante notar que, a princípio, a implementação baseada na versão atual do *middleware* foi pensada como contraponto. Seu propósito é demonstrar que, embora factível, é uma solução muito distante do ideal, se comparada à alternativa. Pode-se argumentar que qualitativamente há uma discrepância perceptível na fase de autoria.

A primeira implementação, além de desnecessariamente complicada do ponto de vista conceitual, traz consigo outros problemas incontornáveis. Primeiramente, coloca como responsabilidade do autor uma configuração muito pouco convencional e tecnicamente complexa, mesmo que se tenha familiaridade com Lua, NCL e NCLua.

Em segundo lugar, força uma suspensão frequente do controle da execução, nos momentos em que esse script verifica as mensagens recebidas via MQTT. Na implementação realizada, isso acontece a cada terço de segundo. Embora seja facilmente configurável, o aumento e a diminuição dessa frequência trazem desvantagens. Se esse intervalo for pequeno demais, a execução da aplicação fica suspensa com muita frequência, afetando diretamente outras funcionalidades. Por outro lado, se esse intervalo for grande demais, expressões faciais podem se perder entre uma consulta e outra.

Como trabalho futuro, é interessante avaliar a experiência de usuários em aplicações com reconhecimento de expressões faciais. É válido acompanhar os avanços na área de reconhecimento facial e avaliar o grau de precisão de algoritmos do estado da arte para aplicações dinâmicas, como programas de televisão.

Um ponto muito importante é a proteção à privacidade do usuário. Faz-se necessário, em trabalhos futuros, estudar e propor soluções a esse respeito, uma vez que, nesta proposta, a aplicação pode usar recursos de reconhecimento facial sem o aval do usuário.

## REFERÊNCIAS

- [1] ABNT. 2011. Digital terrestrial television - Data coding and transmission specification for digital broadcasting - Part 2: Ginga-NCL for fixed and mobile receivers - XML application language for application coding. ABNT NBR 15606-2:2011 standard.
- [2] Fábio Barreto, Raphael S. de Abreu, Eyre Brasil B. Montevecchi, Marina I. P. Josué, Pedro A. Valentim, and Debora C. Muchaluat-Saade. 2020. Extending Ginga-NCL to Specify Multimodal Interactions With Multiple Users. In *Anais do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC.
- [3] Fábio Barreto, Eyre Brasil B. Montevecchi, Raphael Abreu, Joel AF dos Santos, and Debora C. Muchaluat-Saade. 2019. Providing Multimodal User Interaction in NCL. In *Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC, 203–204.
- [4] Fábio Barreto, Eyre Brasil B. Montevecchi, Raphael Abreu, Joel A. F. dos Santos, and Debora C. Muchaluat-Saade. 2019. Providing multi-user in NCL with userAgent and UserProfile. In *Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web* (Florianópolis). SBC, Porto Alegre, RS, Brasil, 205–206. [https://doi.org/10.5753/webmedia\\_estendido.2019.8168](https://doi.org/10.5753/webmedia_estendido.2019.8168)
- [5] Carlos Eduardo CF Batista, Luiz Fernando Gomes Soares, and Guido Lemos de Souza Filho. 2010. Estendendo o uso das classes de dispositivos Ginga-NCL. In *Anais Principais do XVI Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC, 27–34.
- [6] Joseph Bullington. 2005. 'Affective' computing and emotion recognition systems: the future of biometric surveillance?. In *Proceedings of the 2nd annual conference on Information security curriculum development*. 95–99.
- [7] Erik Cambria. 2016. Affective computing and sentiment analysis. *IEEE intelligent systems* 31, 2 (2016), 102–107.
- [8] Julian L. Center Jr and Christopher R. Wren. 2004. Videoconferencing method with tracking of face and dynamic bandwidth allocation. US Patent 6,680,745.
- [9] Jeffrey S. Coffin and Darryl Ingram. 1999. Facial recognition system for security access and identification. US Patent 5,991,429.
- [10] Jeff F. Cohn and Fernando De la Torre. 2015. Automated face analysis for affective computing. (2015).
- [11] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine* 18, 1 (2001), 32–80.
- [12] Rafael Rossi de Mello Brandao, Guido Lemos de Souza Filho, Carlos Eduardo Coelho Freire Batista, and Luiz Fernando Gomes Soares. 2010. Extended features for the Ginga-NCL environment: Introducing the LuaTV API. In *2010 Proceedings of 19th International Conference on Computer Communications and Networks*. IEEE, 1–6.
- [13] Paul Ekman. 1993. Facial expression and emotion. *American psychologist* 48, 4 (1993), 384.
- [14] Paul Ekman. 1999. Facial expressions. *Handbook of cognition and emotion* 16, 301 (1999), e320.
- [15] Nickolaos Fragopanagos and John G. Taylor. 2005. Emotion recognition in human-computer interaction. *Neural Networks* 18, 4 (2005), 389–405.
- [16] Barbara S. Guzak, Hung-Tack Kwan, and Janki Y. Vora. 2011. Multiple sensory channel approach for translating human emotions in a computing environment. US Patent App. 12/540,735.
- [17] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. 2008. MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*. IEEE, 791–798.
- [18] Roberto Ierusalimsky. 2006. *Programming in lua*. Lua.Org Publisher.
- [19] ITU. 2014. Nested Context Language (NCL) and Ginga-NCL. <http://www.itu.int/rec/T-REC-H.761>. ITU-T Recommendation H.761.
- [20] Abhishek Jha. 2007. Class room attendance system using facial recognition system. *The International Journal of Mathematics, Science, Technology and Management* 2, 3 (2007), 4–7.
- [21] Andrea R. Johnson, Matthew J. Johnson, Nico Toutenhoofd, and Clinton L. Fenton. 2014. Assisted photo-tagging with facial recognition models. US Patent 8,861,804.
- [22] Takeo Kanade, Jeffrey F. Cohn, and Yingli Tian. 2000. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE, 46–53.
- [23] E. P. Kukula and S. J. Elliott. 2004. Evaluation of a facial recognition algorithm across three illumination conditions. *IEEE Aerospace and Electronic Systems Magazine* 19, 9 (2004), 19–23.
- [24] Yong Li, Jiabei Zeng, Shiguang Shan, and Xilin Chen. 2018. Occlusion aware facial expression recognition using cnn with attention mechanism. *IEEE Transactions on Image Processing* 28, 5 (2018), 2439–2450.
- [25] Christine L. Lisetti and David E. Rumelhart. 1998. Facial Expression Recognition Using a Neural Network. In *FLAIRS Conference*. 328–332.
- [26] Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. 2016. AFFDEX SDK: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*. 3723–3726.
- [27] David A. Monroe. 2009. Method for incorporating facial recognition technology in a multimedia surveillance system. US Patent 7,634,662.
- [28] Iordanis Mpiperis, Sotiris Malassiotis, and Michael G. Strintzis. 2008. Bilinear models for 3-D face and facial expression recognition. *IEEE Transactions on Information Forensics and Security* 3, 3 (2008), 498–511.
- [29] Rosalind W. Picard. 1999. Affective Computing for HCI. In *HCI (1)*. Citeseer, 829–833.
- [30] Rosalind W. Picard. 2000. *Affective computing*. MIT press.
- [31] Rosalind W. Picard. 2003. Affective computing: challenges. *International Journal of Human-Computer Studies* 59, 1-2 (2003), 55–64.
- [32] Lawrence Sirovich and Michael Kirby. 1987. Low-dimensional procedure for the characterization of human faces. *Josa a* 4, 3 (1987), 519–524.
- [33] Jianhua Tao, Tieniu Tan, and Rosalind W. Picard. 2007. Affective computing and intelligent interaction. In *Second International Conference, ACII*. Springer.
- [34] Matthew A. Turk and Alex P. Pentland. 1991. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*. IEEE Computer Society, 586–587.
- [35] Siyue Xie and Haifeng Hu. 2017. Facial expression recognition with FRR-CNN. *Electronics Letters* 53, 4 (2017), 235–237.
- [36] Alan L. V. Guedes and Simone D. J. Barbosa. 2019. Extending multimedia languages to support multimodal-multiuser interactions. In *Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web* (Florianópolis). SBC, Porto Alegre, RS, Brasil, 5–8. [https://doi.org/10.5753/webmedia\\_estendido.2019.8125](https://doi.org/10.5753/webmedia_estendido.2019.8125)