

Ferramenta de Validação NCL: Aprimorando o processo de desenvolvimento de aplicações de TV Digital

Iago Victor Silva Costa

iago.vsc@discente.ufma.br

Universidade Federal do Maranhão

Li-Chang Shuen

li.chang@ufma.br

Universidade Federal do Maranhão

Daniel de Sousa Moraes

danielmoraes@telemidia.puc-rio.br

TeleMídia Lab - PUC-Rio

Joel André Ferreira dos Santos

jsantos@eic.cefet-rj.br

CEFET/RJ

Débora C. Muchaluat-Saade

debora@midia.com.uff.br

Lab. MídiaCom

Universidade Federal Fluminense

Carlos de Salles Soares Neto

carlos.salles@ufma.br

Universidade Federal do Maranhão

ABSTRACT

The Nested Context Language (NCL) is the declarative language of Ginga, the standard middleware of the Brazilian Digital Terrestrial TV System. From this perspective, the current paper presents an NCL Validation Tool, which presents code diagnostics through error or alert messages, aiming to ease the writing of software for Digital TV. The tool was evaluated using a script of activities with an evaluation scale ranging from 1 (the tool did not help) to 5 (the tool helped completely). Both during the analysis of the script results and the development of the tool, the usefulness of an assistant when authoring NCL applications became evident. Our preliminary results show that the proposed NCL Validation Tool can help productivity and learning of NCL.

KEYWORDS

NCL 4.0, Authoring Tool, Digital TV, NCL Validation

1 INTRODUÇÃO

A linguagem Nested Context Language, ou NCL, linguagem padrão do middleware declarativo do Sistema Brasileiro de TV Digital Terrestre e recomendação ITU-T para Serviços IPTV [9], teve seu uso facilitado ao longo dos anos por meio de ferramentas de autoria. Estas ferramentas envolvem: (i) Linguagens para a criação de *templates*, como o XTemplate [8] e o TAL [10], as quais permitem a descrição de especificações estruturais a partir das quais novos documentos de hipermídia podem ser gerados; (ii) Ferramentas de validação, como o NCL Validator [1], integrado tanto ao *plugin* NCL Eclipse [2] para o ambiente Eclipse quanto ao ambiente NCL Composer [5], que implementam validações léxica, sintática, estrutural, entre outros tipos, no código NCL; (iii) Ambientes de autoria que permitem o desenvolvimento tanto visual quanto textual (Composer, NCL Eclipse, NEXT [12], STEVE [6]) de aplicações NCL e oferecem funcionalidades como validação estrutural de documentos, sugestão de código automático e contextual e coloração de elementos XML e de palavras reservadas da linguagem.

No entanto, estas ferramentas não contam com atualizações recentes e algumas tornaram-se difíceis de acessar e utilizar. Algumas

delas, que podem ser encontrados para *download* de versões pré-compiladas (instaladores/executáveis), não estão mais disponíveis e as opções de instalação/execução que restam podem não ser tão simples para quem não tem costume de usá-las, tornando desejável uma alternativa com acesso simplificado.

Um outro ponto a se levar em conta é o lançamento de novas funcionalidades para a NCL, como a interação multiusuário proposta em [3]. Elaborar um ferramental baseado em uma plataforma mais atual, que dispõe bastante amparo à personalização de seu funcionamento e que recebe constante suporte é interessante não apenas para a criação, mas também para a posterior manutenção de novos recursos.

Além disso, a maior parte dos aparatos que se têm para autoria de apresentações multimídia em NCL é destinada a profissionais de mídia, designers e produtores de conteúdo, isto é, se baseia em abstrações visuais para a criação de documentos. Isto limita o nicho de aplicações, pois a implementação de *features* mais complexas (através do uso de scripts Lua em NCL [13], por exemplo), fica impossibilitada. Sendo assim, é necessária uma nova ferramenta de validação idealmente integrada a uma IDE popular, que permite o emprego conjunto, por exemplo, de extensões NCLua e HTML.

Neste artigo é apresentada uma Ferramenta de Validação NCL, com foco na análise estrutural e comportamental de documentos NCL textuais. Tal ferramenta é um dos requisitos do Ambiente de Autoria de aplicações interativas para TV Digital. Ela está em fase de desenvolvimento e faz parte do projeto de pesquisa TV 3.0, elaborado por um consórcio de nove universidades, Ministério da Ciência e Tecnologia e Rede Nacional de Ensino e Pesquisa.

Por ser parte de um ambiente mais completo, a Ferramenta de Validação NCL por si só não apresenta determinadas funcionalidades presentes, por exemplo, no NCL Eclipse e no NCL Composer, como navegação no documento como uma árvore, execução do documento NCL, entre outros. No entanto, nem todas as *features* relativas à validação textual foram implementadas nesta versão inicial. Algo que ainda não está presente na Ferramenta de Validação NCL, configurando um ponto fraco a ser resolvido, é a sugestão contextual de conteúdo, ou *autocomplete*, a qual também se encontra no NCL Eclipse e no Composer. De modo semelhante, os tipos de validação léxica e sintática não estão implementados. Porém, por sua vez, estas validações não fazem parte do foco da Ferramenta de Validação NCL, sendo adereçadas pelo Schema XML da linguagem, utilizado até o momento como outro componente do Ambiente de Autoria.

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

A Ferramenta de Validação NCL foi avaliada através da aplicação de um roteiro de atividades formatado em um questionário dividido por tópicos, no qual foram respondidas algumas perguntas de múltipla escolha e feitos comentários sobre a ferramenta ao final da realização das tarefas de cada uma dessas etapas.

Tal roteiro de atividades foi criado com base no modelo de questionário SUS (*System Usability Scale*), em que o participante da pesquisa responde cada pergunta em uma escala de 1 (discorda completamente) a 5 (concorda completamente). No contexto desta pesquisa, a escala foi definida no aspecto de o quanto a ferramenta ajudou na identificação de erros no código ao longo de sua escrita: 1 - Nada - a 5 - Completamente.

Por ter envolvido a participação de seres humanos durante o processo de ponderação da ferramenta, foi necessário submeter a pesquisa à avaliação pelo Comitê de Ética da Universidade Federal de Juiz de Fora, por meio da Plataforma Brasil, registrada sob o número 79915024.4.1001.5147.

Este artigo está dividido da seguinte forma: na Seção 2 são exploradas a etapa de levantamento de requisitos e a arquitetura da Ferramenta de Validação NCL; já na Seção 3 são discutidos os resultados obtidos pela aplicação do roteiro de atividades; na Seção 4, são feitas as considerações finais sobre os resultados e a utilidade da ferramenta desenvolvida.

2 FERRAMENTA DE VALIDAÇÃO NCL

A presente seção fornece uma discussão sobre o processo de levantamento de requisitos e descreve a arquitetura proposta para a ferramenta.

2.1 Levantamento de Requisitos

Uma das metas do projeto de TV 3.0 é a disponibilização de um ambiente que facilite o processo de desenvolvimento de aplicações interativas para TV Digital. Tal ambiente, além de outras funcionalidades, traz a validação em tempo real de código NCL, realizada tanto pelo Schema XML que define a linguagem, quanto por uma ferramenta de validação estrutural [1] [11]. A partir de reuniões internas da equipe de Pesquisa e Desenvolvimento (P&D), foram estabelecidos os seguintes tipos de validação estrutural que este módulo deveria executar:

Validação de Atributos - Verifica se atributos relacionados possuem valores corretos

Validação de Referências - Verifica se elemento referenciado é do tipo correto

Validação de Composicionalidade - Verifica se referências são feitas para elementos dentro da composição

Validação de Aninhamento de Composições - Verifica se há loops de aninhamento causados por reuso

Validação de Reuso de elementos - Verifica se há loops de reuso

A fim de atender a estas validações do ambiente de autoria, inicialmente foram levantadas as *tags* que estão envolvidas em cada um desses tipos de validação, bem como as regras e relações associadas a tais *tags*. No decorrer da sondagem por estas *tags*, também fez-se importante entender o funcionamento de algumas áreas funcionais da NCL, visando delimitar a abrangência de cada

Áreas Funcionais	Módulos	Elementos
Structure	Structure	ncl
		head
		body
Layout	Layout	regionBase
		region
Components	Media	media
	Context	context
Interfaces	MediaContentAnchor	area
	CompositeNodeInterface	port
	PropertyAnchor	property
	SwitchInterface	switchPort
Presentation Specification	Descriptor	mapping
		descriptor
		descriptorParam

Tabela 1: Áreas funcionais da NCL 3.0 (1/3)

Linking	Linking	descriptorBase
		bind
		bindParam
		linkParam
Connectors	CausalConnectorFunctionality (agrupa funcionalidades dos módulos: ConnectorCausalExpression; ConnectorCommonPart; ConnectorAssessmentExpression; CausalConnector)	link
		causalConnector
		connectorParam
		simpleCondition
		compoundCondition
		simpleAction
		compoundAction
		assessmentStatement
		attributeAssessment
		valueAssessment
compoundStatement		
Presentation Control	ConnectorBase	connectorBase
		ruleBase
	TestRule	rule
		compositeRule
	TestRuleUse	bindRule
		switch
	ContentControl	defaultComponent
descriptorSwitch		
DescriptorControl	defaultDescriptor	
	Timing	Timing

Tabela 2: Áreas funcionais da NCL 3.0 (2/3)

tipo de validação. Tomou-se como base a tabela apresentada nas Tabelas 1, 2 e 3, retiradas do livro **Programando em NCL 3.0** [14].

Durante todo o processo de estudo da linguagem, foi gerada uma tabela, acessível a partir do link no apêndice, que consistiu em reunir algumas das *tags* da linguagem NCL e seus respectivos atributos para determinar quais estavam envolvidos no que se quis verificar.

Para a primeira versão da ferramenta de validação, apenas os elementos utilizados com mais frequência foram considerados, tendo-se a intenção de acrescentar mais à medida que o ambiente for atualizado. Na tabela A, as células marcadas em cinza indicam elementos que possuem atributos relacionados a alguma validação,

Reuse	Import	importBase
		importDocumentBase
		importNCL
	EntityReuse	
	ExtendedEntityReuse	
Navigational Key	KeyNavigation	
Animation	Animation	
Transition	TransitionBase	transitionBase
Effects	Transition	transition
Meta-Information	Metainformation	meta
		metadata

Tabela 3: Áreas funcionais da NCL 3.0 (3/3)

enquanto as marcadas em amarelo apontam os atributos propriamente ditos. Na coluna “VALIDAÇÃO” estão descritas o que foi validado com respeito ao atributo correspondente.

2.2 Arquitetura da Ferramenta de Validação NCL

A Ferramenta de Validação NCL está sendo desenvolvida como uma extensão para o editor de texto Visual Studio Code [7] e é compatível com as versões 1.82.0 em diante. Extensões permitem adicionar linguagens, depuradores e ferramentas para dar suporte ao trabalho executado.

A ferramenta deve ser instalada no Visual Studio Code como uma extensão, que se comunica com a API de extensões disponibilizada pelo editor para personalizar seu comportamento - neste caso, indicar textual e visualmente erros e/ou ações inesperadas diretamente no código-fonte. O diagrama de componentes ilustrado na Figura 1, descreve a integração da Ferramenta de Validação NCL com a API de extensões fornecida pelo VS Code.

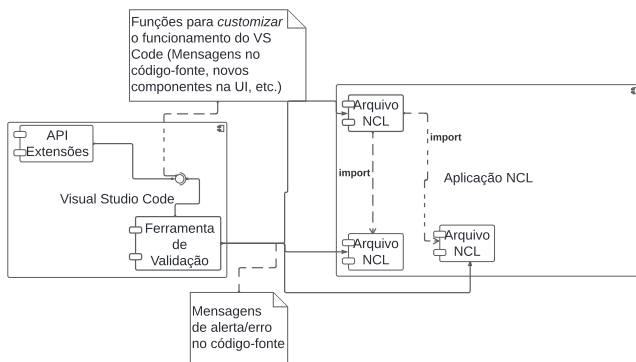


Figura 1: Diagrama de Componentes

Conforme mostra a Figura 2, ao abrir um arquivo do tipo NCL (“arquivo.ncl”, por exemplo) e/ou realizar alguma forma de edição, como apagar e escrever texto ou salvar alterações feitas, a ferramenta guarda uma representação hierárquica de todo o documento - a árvore DOM [4] -, contendo cada tag, suas tags “filhas”, atributos e outros, a partir dos quais verifica o atendimento do código às regras estabelecidas na norma da linguagem NCL.

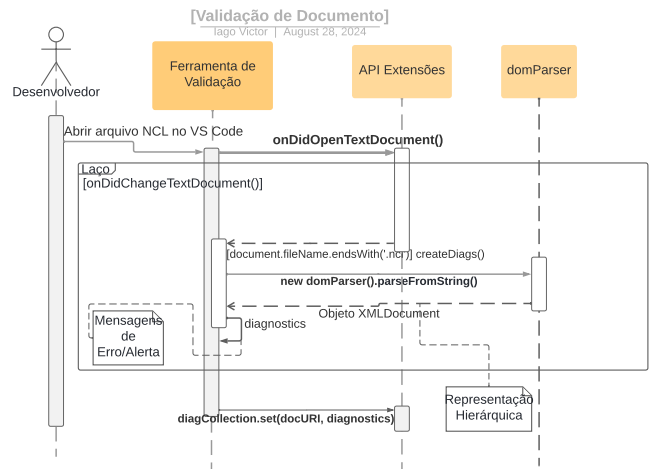


Figura 2: Diagrama de Sequência (Validação de Documento)

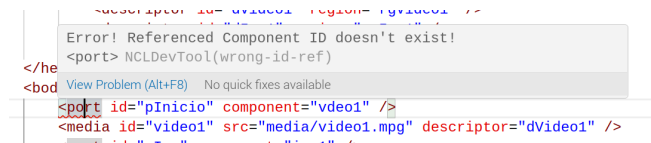


Figura 3: Diagnóstico de Erro

Na Figura 3, a ferramenta aponta que o identificador referenciado pelo atributo *component* da tag `<port>` não existe. Neste exemplo, o atributo deveria referenciar a tag `<media>` imediatamente abaixo, mas o identificador referenciado está escrito sem a letra “i”. Portanto, não havendo nenhuma tag com id “vdeo1”, existe um erro em `<port>`.

3 RESULTADOS

Após a criação e testes do roteiro de avaliação da Ferramenta de Validação NCL, foi solicitada a realização individual das atividades nele presente a alunos de graduação e pós-graduação em cursos da área de Computação de diferentes universidades que soubessem codificar utilizando a linguagem NCL em um nível ao menos básico - criação de aplicações com comportamentos definidos por elos.

Foram obtidas sete respostas ao roteiro, as quais configuraram o quadro de avaliações da Figura 4, onde cada seção é representada por uma barra.

Note que, para o participante 6, a avaliação da seção sobre Reuso em NCL é zero, opção referente à impossibilidade de completar todos os passos. Tal opção foi adicionada devido à abrangência das validações definidas durante o projeto da ferramenta, que levou a um certo salto de dificuldade nas tarefas a serem executadas. Uma situação semelhante ocorreu com o participante 7, com a diferença de que, além da questão de conhecimento da linguagem, também há o fato de que ele utilizou o *OSS Code*, uma versão de código aberto do *Visual Studio Code* sem código proprietário da *Microsoft*, que possivelmente causou erros internos durante a execução da extensão da ferramenta.

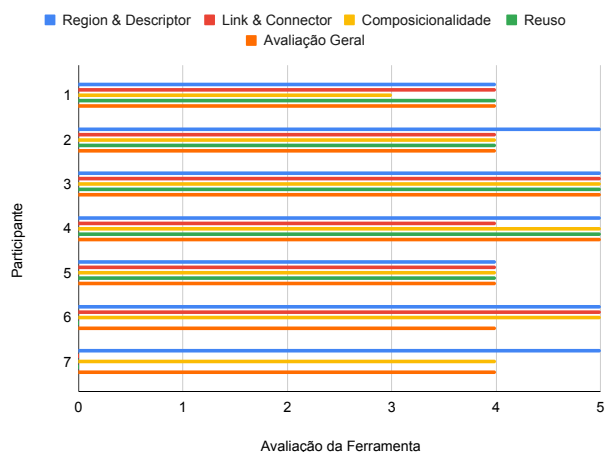


Figura 4: Avaliações em cada Seção por Participante

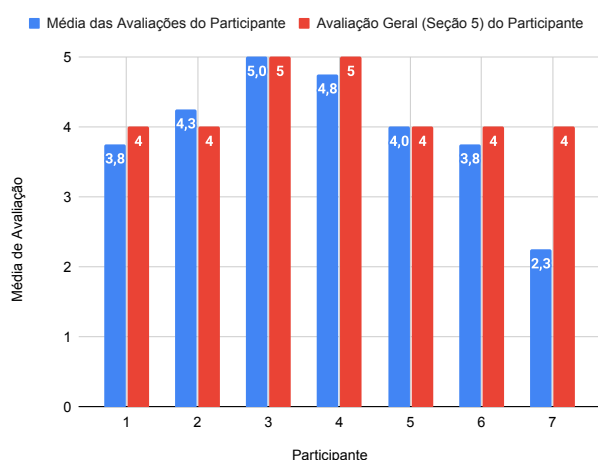


Figura 5: Média de Avaliação por Participante

Apesar destes impasses, a maioria das avaliações foi de pontuação 4 - a ferramenta ajudou muito na identificação de erros (maioria deles) - e 5 - todos os erros foram identificados e corrigidos devido à ferramenta -, apontando para a satisfação de se ter um validador que assinala falhas em tempo real.

Como se pode observar na Figura 5, não houve disparidade entre as médias das avaliações (primeiras 4 seções) de cada participante e suas respectivas avaliações gerais (dadas na seção 5 do roteiro) para a maioria dos casos, o que demonstra constância tanto do impacto da Ferramenta de Validação ao longo das áreas funcionais consideradas nesta primeira versão quanto das respostas dos voluntários deste estudo, refletindo sua veracidade.

Em relação ao caso do participante 7, a média de suas avaliações foi baixa devido à não completude dos passos em 2 seções, derivando daí avaliação 0 em ambas. No entanto, considerando-se apenas as

outras duas seções, que foram completamente feitas, a média passa a ser de 4.5/5, bem mais próxima à classificação geral.

Segundo a Figura 6, apesar da dificuldade acentuada dos passos do roteiro a partir da seção sobre Composicionalidade (seção 3), a menor média que a ferramenta atingiu foi de 3.1/5, enquanto as outras se aproximaram mais de 4.

Por fim, apesar das avaliações quantitativas de média alta, dois problemas pontuais foram explicitados por meio de comentários feitos pelos participantes: (i) É acusado erro na importação de documentos caso o valor do atributo *documentURI* contenha os caracteres *"/*", usados comumente para se referir ao diretório onde o documento aberto se encontra, o que causa a notificação incorreta de erro tanto na importação quanto nas referências aos elementos importados, mesmo que estejam corretas; (ii) Os erros marcados não são exibidos na aba *Problems* do VS Code, sendo possível visualizar suas descrições apenas ao posicionar o cursor em cima de sua indicação visual.

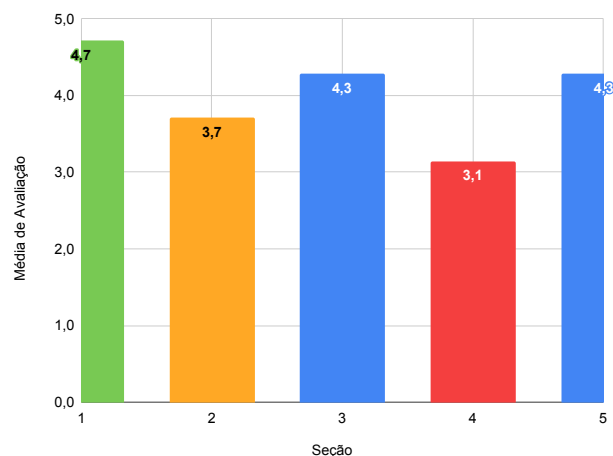


Figura 6: Média de Avaliação por Seção

4 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi avaliar a utilidade de um validador para a linguagem NCL, que facilita e acelera a criação de aplicações interativas de Televisão. Sendo parte do Ambiente de Autoria, tecnologia proposta a fim de suprir um dos requisitos do projeto TV 3.0, este validador, referido como Ferramenta de Validação NCL, poderá impulsionar o uso da linguagem na produção de software voltado à TV Digital brasileira.

Tanto no decorrer da elaboração deste trabalho, quando outros integrantes das equipes de Pesquisa & Desenvolvimento do projeto TV 3.0 puderam utilizar a ferramenta, quanto no momento da análise das respostas do roteiro, observou-se que ter acesso simples e rápido a uma ferramenta que norteie o desenvolvedor durante a escrita de cada parte do programa é de grande valia.

Os resultados apresentados neste trabalho são qualitativos e compostos de uma pequena amostra de voluntários. Não é tão simples alcançar um número maior de voluntários, dadas as especificidades necessárias para tal. No entanto, como trabalho futuro espera-se

confirmar os resultados preliminares apresentados neste artigo com um grupo maior de usuários.

Outro desdobramento possível é fazer uma avaliação sobre os benefícios na integração da ferramenta com extensões NCLua e HTML da IDE Microsoft Visual Studio. Neste estudo futuro pretende-se medir o impacto de se ter uma IDE integrada que permita a criação de aplicações de TV digital que sejam tanto declarativas quanto imperativas.

AGRADECIMENTOS

Os autores agradecem ao Ministério da Comunicações, Fórum SBTVD e RNP pelo suporte financeiro para este trabalho.

REFERÊNCIAS

- [1] Eduardo C Araújo, Roberto G de A Azevedo, and Carlos de SS Neto. 2008. NCL-validator: um processo para validação sintática e semântica de documentos multimídia NCL. *II Jornada de Informática do Maranhão-São Luís, Brasil* (2008).
- [2] RG de A Azevedo, Mário M Teixeira, and CS Soares Neto. 2009. Ncl eclipse: Ambiente integrado para o desenvolvimento de aplicações para tv digital interativa em nested context language. In *SBRC: Simpósio Brasileiro de Redes de Computadores*.
- [3] Fabio Barreto, Raphael Abreu, and Débora Muchalut-Saade. 2023. TV 3.0: Interação Multiusuário para TV Digital Aberta com NCL 4.0. In *Anais Estendidos do XXIX Simpósio Brasileiro de Sistemas Multimídia e Web* (Ribeirão Preto/SP). SBC, Porto Alegre, RS, Brasil, 179–184. https://doi.org/10.5753/webmedia_estendido.2023.236162
- [4] Sabin-Corneliu Buraga. [n. d.]. Document Object Model (DOM)-o cale de acces. ([n. d.]).
- [5] Rodrigo Laiola Guimarães, RMR Costa, and Luiz Fernando Gomes Soares. 2007. Composer: Ambiente de autoria de aplicações declarativas para TV Digital Interativa. *WEBMEDIA: Brazilian Symposion on Multimedia and the Web* (2007).
- [6] Douglas Paulo de Mattos and Débora Christina Muchalut Saade. 2016. STEVE: Spatial-Temporal View Editor for Authoring Hypermedia Documents. In *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web* (Teresina, Piaui State, Brazil) (*Webmedia '16*). Association for Computing Machinery, New York, NY, USA, 63–70. <https://doi.org/10.1145/2976796.2976865>
- [7] Microsoft. 2024. *Visual Studio Code*. <https://code.visualstudio.com/>
- [8] Débora Christina Muchalut-Saade and Luiz Fernando Gomes Soares. 2002. XConertor and XTemplate: improving the expressiveness and reuse in web authoring languages. *New review of hypermedia and multimedia* 8, 1 (2002), 139–169.
- [9] Carlos de Salles Soares Neto and Luiz Fernando Gomes Soares. 2009. Reuse and imports in nested context language. In *Proceedings of the XV Brazilian Symposium on Multimedia and the Web*. 1–8.
- [10] Carlos de Salles Soares Neto, Luiz Fernando Gomes Soares, and Clarisse Sieckenius de Souza. 2012. TAL-Template Authoring Language. *J. Braz. Comput. Soc.* 18, 3 (2012), 185–199.
- [11] José Rios Cerqueira Neto, Rodrigo Costa Mesquita Santos, Carlos de Salles Soares Neto, and Mário Meireles Teixeira. 2011. Método de validação estrutural e contextual de documentos ncl. In *Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*. SBC, 1–8.
- [12] Douglas Paulo de Mattos, Júlia Varanda da Silva, and Débora Christina Muchalut-Saade. 2013. NEXT: graphical editor for authoring NCL documents supporting composite templates. In *Proceedings of the 11th European Conference on Interactive TV and Video* (Como, Italy) (*EuroITV '13*). Association for Computing Machinery, New York, NY, USA, 89–98. <https://doi.org/10.1145/2465958.2465964>
- [13] Francisco Sant'Anna, Renato Cerqueira, and Luiz Fernando Gomes Soares. 2008. NCLua: objetos imperativos lua na linguagem declarativa NCL. In *Proceedings of the 14th Brazilian Symposium on Multimedia and the Web* (Vila Velha, Brazil) (*WebMedia '08*). Association for Computing Machinery, New York, NY, USA, 83–90. <https://doi.org/10.1145/1666091.1666107>
- [14] LUIZ FERNANDO GOMES SOARES Soares. 2009. *Programando em NCL 3.0: desenvolvimento de aplicações para middleware Ginga: TV digital e Web*. Elsevier.

A TABELA DE RELAÇÃO ELEMENTO NCL - ATRIBUTOS - VALIDAÇÃO

Link para a tabela: <https://shorturl.at/xcEIw>