

LILIAN COELHO DE FREITAS
(ORGANIZADORA)

Collection:

APPLIED COMPUTER ENGINEERING

Atena
Editora
Ano 2022

LILIAN COELHO DE FREITAS
(ORGANIZADORA)

Collection:

APPLIED COMPUTER ENGINEERING

Editora chefe

Profª Drª Antonella Carvalho de Oliveira

Editora executiva

Natalia Oliveira

Assistente editorial

Flávia Roberta Barão

Bibliotecária

Janaina Ramos

Projeto gráfico

Camila Alves de Cremo

Daphynny Pamplona

Gabriel Motomu Teshima

Luiza Alves Batista

Natália Sandrini de Azevedo

Imagens da capa

iStock

Edição de arte

Luiza Alves Batista

2022 by Atena Editora

Copyright © Atena Editora

Copyright do texto © 2022 Os autores

Copyright da edição © 2022 Atena Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.

Open access publication by Atena Editora



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição *Creative Commons*. Atribuição-Não-Comercial-Não-Derivativos 4.0 Internacional (CC BY-NC-ND 4.0).

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores, inclusive não representam necessariamente a posição oficial da Atena Editora. Permitido o *download* da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

Conselho Editorial**Ciências Exatas e da Terra e Engenharias**

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto

Profª Drª Alana Maria Cerqueira de Oliveira – Instituto Federal do Acre

Profª Drª Ana Grasielle Dionísio Corrêa – Universidade Presbiteriana Mackenzie

Profª Drª Ana Paula Florêncio Aires – Universidade de Trás-os-Montes e Alto Douro

Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás

Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná



Prof. Dr. Cleiseano Emanuel da Silva Paniagua – Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Profª Drª Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará
Profª Dra. Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho
Prof. Dr. Juliano Bitencourt Campos – Universidade do Extremo Sul Catarinense
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande
Profª Drª Luciana do Nascimento Mendes – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Prof. Dr. Marcelo Marques – Universidade Estadual de Maringá
Prof. Dr. Marco Aurélio Kistemann Junior – Universidade Federal de Juiz de Fora
Prof. Dr. Miguel Adriano Inácio – Instituto Nacional de Pesquisas Espaciais
Profª Drª Neiva Maria de Almeida – Universidade Federal da Paraíba
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Profª Drª Priscila Tessmer Scaglioni – Universidade Federal de Pelotas
Prof. Dr. Sidney Gonçalo de Lima – Universidade Federal do Piauí
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista



Diagramação: Camila Alves de Cremo
Correção: Yaidy Paola Martinez
Indexação: Amanda Kelly da Costa Veiga
Revisão: Os autores
Organizadora: Lilian Coelho de Freitas

Dados Internacionais de Catalogação na Publicação (CIP)

C697 Collection: applied computer engineering / Organizadora
Lilian Coelho de Freitas. – Ponta Grossa - PR: Atena,
2022.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-65-5983-859-2

DOI: <https://doi.org/10.22533/at.ed.592222801>

1. Computer engineering. I. Freitas, Lilian Coelho de
(Organizadora). II. Título.

CDD 621.39

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora

Ponta Grossa – Paraná – Brasil

Telefone: +55 (42) 3323-5493

www.atenaeditora.com.br

contato@atenaeditora.com.br



DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.



DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access*, *desta forma* não os comercializa em seu site, sites parceiros, plataformas de *e-commerce*, ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.



APRESENTAÇÃO

Atena Editora is honored to present the e-book entitled "*Collection: Applied Computer Engineering*". This volume presents 17 chapters about applications of computer engineering in industrial automation, robotics, data science, information security, neuromarketing, speech development in children, among others.

We want to take this moment to thank all of our authors for entrusting us with their discoveries. We are also grateful to the reviewers and readers who have contributed to the success of our books.

Enjoy your reading.

Lilian Coelho de Freitas

SUMÁRIO

CAPÍTULO 1..... 1

ALIMENTADOR AUTOMÁTICO DE PET UTILIZANDO A PLATAFORMA ARDUÍNO

Márcio Valério de Oliveira Favacho

Vivian da Silva Lobato

Raphael Saraiva de Sousa

Alberto Cauã Trindade da Silva

Denise Nascimento Cardoso

Jamilly da Silva Dias

Jéssica Ferreira e Ferreira

Pedro Afonso Alcântara Negrão

Rízia de Cássia da Fonseca Pereira

Ruam Melo dos Santos

Weliton Quaresma Ferreira

 <https://doi.org/10.22533/at.ed.5922228011>

CAPÍTULO 2..... 14

ANÁLISE DE AGRUPAMENTO PARA APRIMORAR A EXTRAÇÃO AUTOMÁTICA DE DEMONSTRATIVOS FINANCEIROS COM ESTUDO DE ESCALABILIDADE

Igor Raphael Magollo

Gabriel Olivato

Victor Vieira Ferraz

Murilo Coelho Naldi

 <https://doi.org/10.22533/at.ed.5922228012>

CAPÍTULO 3..... 32

AVALIANDO A USABILIDADE DE APLICAÇÕES VOLTADAS PARA A COMUNICAÇÃO DE CRIANÇAS COM TEA

Joêmia Leilane Gomes de Medeiros

Welliana Benevides Ramalho

Edinadja Mayara de Macedo

 <https://doi.org/10.22533/at.ed.5922228013>

CAPÍTULO 4..... 47

CONTROLE E MONITORAMENTO AUTOMATIZADO DOS FATORES LIMNOLÓGICOS IDEAIS PARA LARVICULTURA DO PTEROPHYLLUM SCALARE (ACARÁ BANDEIRA) UTILIZANDO TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL

Raphael Saraiva de Sousa

Otávio Noura Teixeira

Augusto César Paes de Souza

Márcio Valério de Oliveira Favacho

Renato Hidaka Torres

 <https://doi.org/10.22533/at.ed.5922228014>

CAPÍTULO 5..... 63

GESTIÓN DE RIESGOS Y CONTINUIDAD DEL NEGOCIO SOBRE LA SEGURIDAD

INFORMÁTICA EN EL SECTOR RETAIL EN MÉXICO

José Eduardo Mendoza Macias

Emigdio Larios Gómez

 <https://doi.org/10.22533/at.ed.5922228015>

CAPÍTULO 6..... 73

IAÇÁ – OTIMIZAÇÃO DO PROCESSO DE EXTRAÇÃO DA POLPA DE AÇÁ UTILIZANDO A PLATAFORMA ARDUÍNO

Márcio Valério de Oliveira Favacho

Vivian da Silva Lobato

Adenildo da Conceição Silva da Silva

Ana Flavia Dias da Silva

Ian Castro Marinho da Silva

Leonan Gustavo Silva Rodrigues

Lilian Raquel de Campos Cardoso

Marily Luciene Pantoja Costa

Nayra Pereira Ferreira

Paulo Vitor Melo Amaral Ferreira

Rodrigo Figueiró Santana

 <https://doi.org/10.22533/at.ed.5922228016>

CAPÍTULO 7..... 84

LINGUAGEM DE DOMÍNIO ESPECÍFICO PARA A AUTORIA DE APLICAÇÕES PARA TV DIGITAL

Lucas de Macedo Terças

Daniel de Sousa Moraes

Carlos de Salles Soares Neto

 <https://doi.org/10.22533/at.ed.5922228017>

CAPÍTULO 8..... 95

NEUROMARKETING APLICADO AO EMOCIONAL BRANDING

Maiara Bettu

Vanessa Angélica Balestrin

 <https://doi.org/10.22533/at.ed.5922228018>

CAPÍTULO 9..... 111

PROPOSTA DE METAMODELOS DE GEOVISUALIZAÇÃO COM RECURSOS ADAPTÁVEIS

Ítalo Moreira Silva

Alexandre Carvalho Silva

Camilo de Lellis Barreto Junior

Diogo Aparecido Cavalcante de Lima

 <https://doi.org/10.22533/at.ed.5922228019>

CAPÍTULO 10..... 116

SISTEMA INTEGRAL AUTOMATIZADO DE SEGUIMIENTO DE EGRESADOS Y

EMPLEADORES

Leonor Angeles Hernández
Mónica Leticia Acosta Miranda
Daniel Domínguez Estudillo
Edi Ray Zavaleta Olea
José Arnulfo Corona Calvario

 <https://doi.org/10.22533/at.ed.59222280110>

CAPÍTULO 11..... 126

STRENGTH PREDICTION OF ADHESIVELY-BONDED JOINTS WITH COHESIVE LAWS ESTIMATED BY DIGITAL IMAGE CORRELATION

Ulisses Tiago Ferreira Carvalho
Raul Duarte Salgueiral Gomes Campilho

 <https://doi.org/10.22533/at.ed.59222280111>

CAPÍTULO 12..... 140

TAGARELAPP: PROTÓTIPO DE INTERFACE CENTRADO NA USABILIDADE PARA O DESENVOLVIMENTO DA FALA E COMUNICAÇÃO DE CRIANÇAS COM TEA

Joêmia Leilane Gomes de Medeiros
Welliana Benevides Ramalho
Edinadja Mayara de Macedo

 <https://doi.org/10.22533/at.ed.59222280112>

CAPÍTULO 13..... 152

ESTRATEGIA DE MIGRACIÓN DE UN SISTEMA LEGADO UTILIZANDO LA METODOLOGÍA “CHICKEN LITTLE” APLICADA AL SISTEMA DE BEDELÍAS DE LA UNIVERSIDAD DE LA REPÚBLICA DE URUGUAY

Cristina González
Mariela De León

 <https://doi.org/10.22533/at.ed.59222280113>

CAPÍTULO 14..... 169

INTRODUÇÃO A ANÁLISE FORENSE COMPUTACIONAL: DETECTANDO ROOTKITS EM AMBIENTE WINDOWS

Thiago Giroto Milani
Ricardo Slavov

 <https://doi.org/10.22533/at.ed.59222280114>

CAPÍTULO 15..... 191

USO DAS TICS COMO METODO PARA ELABORAR TRABALHO RECEPCIONAL E PLATAFORMA PARA A AUTOMATIZAÇÃO DE FORMATOS DE ESTADIAS

Eloína Herrera Rodríguez
Sonia López Rodríguez
Claudia Galicia Solís

 <https://doi.org/10.22533/at.ed.59222280115>

CAPÍTULO 16	209
NARRATIVAS ACADÊMICAS EM PESQUISA: MÁQUINAS DE GUERRA VIRTUAIS	
Angeli Rose	
 https://doi.org/10.22533/at.ed.59222280116	
CAPÍTULO 17	218
OPTIMIZATION BASED OUTPUT FEEDBACK CONTROL DESIGN IN DESCRIPTOR SYSTEMS	
Elmer Rolando Llanos Villarreal	
Maxwell Cavalcante Jácome	
Edpo Rodrigues de Morais	
João Victor de Queiroz	
Walter Martins Rodrigues	
 https://doi.org/10.22533/at.ed.59222280117	
SOBRE A ORGANIZADORA	225
ÍNDICE REMISSIVO	226

LINGUAGEM DE DOMÍNIO ESPECÍFICO PARA A AUTORIA DE APLICAÇÕES PARA TV DIGITAL

Data de aceite: 10/01/2022

Data de submissão: 19/10/2021

Lucas de Macedo Terças

UFMA - Universidade Federal do Maranhão
São Luís, Maranhão
<http://lattes.cnpq.br/0407755570356994>

Daniel de Sousa Moraes

UFMA - Universidade Federal do Maranhão
São Luís, Maranhão
<http://lattes.cnpq.br/5881729654998250>

Carlos de Salles Soares Neto

UFMA - Universidade Federal do Maranhão
São Luís, Maranhão
<http://lattes.cnpq.br/1512846862093142>

RESUMO: O Sistema Brasileiro de TV Digital Terrestre (SBTVD-T) utiliza a linguagem NCL para a autoria de aplicações multimídia. Embora tal linguagem tenha sido criada com o intuito de ser fácil de ser entendida por produtores de conteúdo dessas aplicações, o fato dela ser baseada em XML introduz um certo nível de verbosidade que acaba degradando a usabilidade da linguagem e atrasando o processo de desenvolvimento de desenvolvedores de software mais experientes. Esse trabalho apresenta a sNCL (Simpler NCL), uma linguagem de domínio específico focada na redução de verbosidade na construção de aplicações para o middleware Ginga-NCL e aumento de usabilidade comparado com as soluções atuais. A abordagem da sNCL não atua como um substituto de NCL, onde

documentos sNCL são compilados para NCL e assim tocados no player. O trabalho também apresenta o desenvolvimento do compilador da linguagem, assim como análises e comparações da usabilidade da sNCL comparadas com NCL.

PALAVRAS-CHAVE: TV Digital, Compilador, PEG.

DOMAIN SPECIFIC LANGUAGE FOR AUTHORIZING DIGITAL TV APPLICATIONS

ABSTRACT: The Brazilian Terrestrial Digital TV System (SBTVD-T) uses the NCL language for multimedia application's authoring. Although such language has been designed to be easy to understand by producers of such applications, the fact of being XML based introduces a reasonable level of verbosity that ends up degrading the usability of the language and delaying more experienced software developers. This paper presents sNCL (simpler NCL) a domain specific language (DSL) focused on the reduction of verbosity in the construction of NCL documents. The sNCL approach does not act as a replacement to the use of NCL, but as an alternative language if the author wants a simpler and more concise language, given that sNCL documents are compiled to NCL and then played in the Ginga-NCL player. This paper also presents the development of the compiler of the language, as well as analysis and comparisons with NCL based on the CDN framework.

KEYWORDS: Digital TV, Compilers, PEG.

1 | INTRODUÇÃO

Sistemas de autoria multimídia têm sido discutidos na área acadêmica há anos. Um desafio tem sido prover formas de tornar o processo de criação de apresentações com diferentes tipos de conteúdo e seus relacionamentos o mais simples e eficaz possível assim como acessível a usuários experientes ou não na área.

NCL (Nested Context Language) é uma linguagem declarativa de alto nível de abstração, baseada em XML e no modelo NCM (Nested Context Model), usada na autoria de aplicações multimídia para TV Digital no Brasil e também recomendação ITU-T para serviços IPTV e padrão ISDB-TB do Sistema Nipo-Brasileiro de TV Digital Terrestre. Esse tipo de aplicação tem como um dos seus autores alvos os produtores de conteúdo e profissionais das emissoras de TV. Por isso, há a necessidade de que a linguagem usada para autoria seja de fácil leitura e entendimento por esses profissionais que não necessariamente tem conhecimento em desenvolvimento de aplicações.

XML foi pensada para facilitar a abstração e especificação de novas linguagens, o que significa que linguagens baseadas nela são verbosas, por que ela apresenta um escopo muito grande, diminuindo a usabilidade da mesma. Embora isso facilite a criação de aplicações por quem não tem conhecimento de programação, como programadores-visuais e desenvolvedores de animações, à medida que o documento da aplicação cresce ele fica muito confuso e difícil de ler. Além disso, há o fato que programadores experientes produzindo aplicações para TV Digital não precisam de tal verbosidade e isso até atrapalha sua produtividade, geralmente preferindo linguagens mais compactas e simples.

O reúso em NCL foi implementado para possibilitar a reutilização de código tanto dentro da aplicação quanto por aplicações diferentes e até por bibliotecas externas, o que diminui a probabilidade de erros se o código reusado for bem testado e diminui também o tempo de desenvolvimento da aplicação. Embora a técnica de reúso seja geralmente implementada em linguagens imperativas, ela foi usada em NCL, uma linguagem declarativa, pelo fato de que na criação de aplicações para TV Digital existe a possibilidade de mídias diferentes terem características de apresentação iguais. Exemplos disso são o como ou onde as mídias serão mostradas.

Tendo como objetivo criar uma linguagem mais simples e menos verbosa para desenvolvedores, este trabalho apresenta a linguagem declarativa de domínio específico sNCL (*simpler* NCL), em que o design da mesma foi feito baseado no framework CDN (Cognitive Dimensions of Notations) assim como questionários com usuários com experiência em NCL para validar a usabilidade da mesma.

2 | FUNDAMENTAÇÃO TEÓRICA

Nessa seção será apresentado um resumo de trabalhos relacionados a diminuição de verbosidade ou aumento de produtividade em autoria de aplicações multimídia para o

sistema do NCL, assim como a fundamentação que foi usada para iniciar o desenvolvimento da sintaxe da nova linguagem.

2.1 Trabalhos relacionados

Existem vários trabalhos comprometidos com a redução da verbosidade da NCL, que variam de *scripts* ou sistemas de *templates* para a autoria até a sugestão de um novo perfil NCL, chamado NCL Raw, que visa ser uma linguagem intermediária entre o usuário e o *middleware*, removendo açúcares sintáticos e entidades do NCL. Cada abordagem sugere meios de tornar a autoria de aplicações multimídia mais suave aos autores alvo.

Em TAL, os autores propõem uma linguagem para autoria de *templates* de aplicações hipermídia, onde a criação de uma aplicação passa a ter duas etapas: a autoria dos *templates*, que são documentos incompletos que descrevem aplicações deixando lacunas; e a criação da aplicação final, onde tais lacunas são preenchidas. Apresenta-se a separação dos papéis envolvidos na autoria, onde se tem o autor de *templates* e o autor final, que produz uma aplicação a partir dos *templates* disponíveis. Com essa divisão, a aplicação em desenvolvimento possui diferentes arquivos com tamanhos reduzidos.

O trabalho em Luar introduz um sistema mais robusto para processamento de *templates*, onde se tem dois processadores: um responsável pelo processamento dos *templates* e geração de uma biblioteca dos mesmos (chamados de documentos Luar), que é compartilhada entre os desenvolvedores; e um segundo chamado processador de aplicação, responsável por interpretar os *templates* e os enviar ao primeiro processador. A biblioteca também permite a inserção de trechos de código Lua no documento NCL, transformando-os em um documento luar.

Esses dois trabalhos se relacionam ao proposto neste artigo por aplicar técnicas de reuso.

No trabalho Lua2NCL, usa-se a linguagem Lua para criar um documento que passa por rotinas responsáveis pela geração do arquivo NCL final. Nesse projeto são usadas tabelas Lua para representar os elementos do documento NCL, onde cada elemento é representado como uma classe.

Ao instanciar um objeto de umas das classes, não há a necessidade de repetição dos termos, como acontece com as *tags* XML. Embora haja uma redução nos documentos de aplicações, o uso de tabelas Lua ainda possui uma verbosidade considerável quando se tem muitos aninhamentos de tabelas. Isso porque a sintaxe de Lua para tabelas exige o uso de muitos símbolos, como chaves e colchetes, o que pode ainda não ser uma boa opção para simplificar o processo de criação de aplicações.

Já em JNS, o trabalho propõe uma forma de autoria alternativa, chamada JNS (*JSON NCL Script*) em que o desenvolvedor cria um documento JSON e o compila para NCL. O JNS também tem como objetivo reduzir a verbosidade. Tendo como alvo o usuário final, sejam estes desenvolvedores e produtores de conteúdo. Oferece-se uma sintaxe

mais limpa, tornando o desenvolvimento de aplicações mais rápido e fácil, provendo uma alternativa a desenvolvedores que não gostam do estilo de um documento XML. No entanto, essa alternativa requer o conhecimento de JSON.

Em ambos os trabalhos, Lua2NCL e JNS utilizam-se de estruturas de linguagens já existentes, Lua e JSON respectivamente, para a construção dos documentos, o que exige o conhecimento prévio das linguagens base.

No entanto, são linguagens para diferentes perfis, já que JSON é bastante usada para aplicações web, sendo uma linguagem para modelagem de dados, enquanto Lua é uma linguagem imperativa, além de outras utilizações, vem sendo muito usada nas aplicações DTV. A linguagem proposta tem sintaxe própria e reduz o uso de símbolos como chaves e dois pontos, que são ao bastantes usados nas abordagens citadas.

O trabalho do NCL Raw propõe um novo perfil para NCL, que remove as estruturas de reuso, deixando somente *links*, *context*, *port*, *média* e *propriedades*.

O projeto NCL Raw foi proposto a fim de melhorar o processo de conversão do documento NCL para o modelo do player Ginga-NCL, e, para isso, o perfil remove os açúcares sintáticos e as possibilidades de reuso do NCL, para implementar uma linguagem intermediária, mas que ainda assim é fácil de entender para pessoas que não tem experiência com programação.

O trabalho apresentado neste artigo se baseia no citado logo acima para um uso mínimo de estruturas sem afetar a semântica das aplicações e promovendo o reuso.

2.2 CDN

Para se iniciar o design da sintaxe da linguagem, uma análise da NCL foi feita usando o framework CDN. A aplicação do CDN remarca os princípios para melhorar notações em linguagens de programação, avaliando a sua usabilidade ou ajudando a projetar uma nova. No caso do trabalho atual, o framework foi utilizado para validar a proposta de sintaxe da linguagem com usuários com experiência em NCL e em autoria de aplicações multimídia, estabelecendo assim diretrizes para o design e melhorias da linguagem.

Pesquisas anteriores mostram que o CDN provê informações importantes sobre como interpretar o impacto e o significado da semântica e sintaxe de linguagens de programação sobre a percepção humana.

Além desses fatores, a CDN também leva em conta todo o ambiente ao redor da notação, em linguagens de programação isso se traduz em ambientes de desenvolvimento, IDEs, e o próprio compilador da linguagem, se ele apresenta para o desenvolvedor erros claros e significativos, por exemplo, e outras aplicações que possam ajudar no desenvolvimento da aplicação.

Thomas Green originalmente definiu 12 dimensões cognitivas:

1. Gradientes de Abstração
2. Proximidade de Mapeamento

3. Difusão
4. Facilidade a Erros
5. Operações Mentais Díficeis
6. Dependências Escondidas
7. Consistência
8. Compromisso Prematuro
9. Avaliação Progressiva
10. Notação Secundária
11. Viscosidade
12. Visibilidade

Áreas Funcionais de NCL	Elementos NCL	Dimensão CDN
Apresentação	<region> <transition> <descriptor> <context> <media>	- Visibilidade - Viscosidade - Dependências Escondidas
Relacionamentos	<connector> <link>	- Dependências Escondidas - Visibilidade - Compromisso Prematuro - Viscosidade - Facilidade de Erros

Tabela 1: Demonstração dos resultados da análise de NCL com o CDN

Após essa análise, algumas decisões de design foram tomadas em sNCL, como a remoção do cabeçalho e do corpo, pois a separação deles trazia ocorrências de dependências escondidas no documento, assim como atrapalhava a visibilidade. Remoção dos elementos conector e elo, em vez disso criando um novo elemento que representa tanto o modelo do relacionamento quanto os elementos que o compõem, assim eliminando um dos problemas que havia com esses elementos em NCL que era que eles são interdependentes, porém ficavam em lugares distantes no documento.

3 | SNCL

Este trabalho apresenta a criação da sNCL, uma linguagem de domínio específico declarativa, voltada para o desenvolvimento de aplicações multimídia para o SBTVD-T, visando diminuir a verbosidade do documento final, e fazer a autoria de tais aplicações por desenvolvedores mais fácil.

A sNCL planeja trazer um modo mais simples na autoria de aplicações para o Ginga-NCL. As aplicações escritas usando a sNCL são compiladas para a linguagem final

usada no middleware, a NCL, usando como parser a biblioteca Lua Parsing Expression Grammar (LPeg) e assim fornecer aos desenvolvedores que não querem usar o estilo de um documento XML, preferindo uma sintaxe mais parecida com as línguas a que estão familiarizados.

A linguagem proposta neste trabalho sugere o uso de uma sintaxe parecida com a da linguagem imperativa Lua, já que Lua também é usada como uma linguagem de extensão e script para aplicações de TV Digital, o NCLua. Portanto, o uso da sintaxe semelhante proporciona aos desenvolvedores uma familiaridade com a nova linguagem, levando em conta os resultados obtidos pelo CDN.

O compilador de sNCL é feito em Lua, onde a gramática da linguagem é representada por uma PEG (Parsing Expression Grammar), um tipo de gramática introduzida por Bryan Ford. Diferentemente das gramáticas Livres de Contexto, onde uma expressão pode resultar em mais de uma árvore, a PEG não possui ambiguidade, cada cadeia analisada possui apenas uma árvore que a representa. O LPeg é uma biblioteca Lua que baseia-se nesse conceito para auxiliar na análise de PEGs, por isso será usada para fazer a construção do compilador da sNCL. E os elementos do documento sNCL são representados por uma tabela de símbolos indexada pelos IDs dos elementos.

3.1 Elementos nativos de NCL

A sNCL usa os elementos de NCL, como mídia, âncora, contexto, e seus atributos como palavras reservadas da língua, que simulam um tipo de dado, como int, float. A figura 1 apresenta um resumo dos elementos de apresentação de sNCL, uma região, uma mídia e as âncoras filhas das mídias.

```
1 region regionTv
2   width = "100%"
3   height = "100%"
4   zIndex = "10"
5 end
6 media videoIntroducao
7   region = regionTv
8   source = "../videos/videoIntroducao.mp4"
9   area area01
10    begin = "5s"
11    end = "10s"
12  end
13  area area02
14    begin = "10s"
15    end = "14s"
16  end
17 end
```

Figura 1: Elementos de apresentação em sNCL.

Esses elementos em NCL seriam declarados como na figura 2 abaixo. Nota-se a remoção do elemento descritor, já que ele só servia como uma ligação entre o elemento região e o elemento da mídia, com nenhuma propriedade sendo declarada nele.

```
1 <head>
2   <regionBase>
3     <region id="regionTv" width="100%" height="100%" zIndex="10" />
4   </regionBase>
5   <descriptorBase>
6     <descriptor id="descriptorTv" region="regionTv" />
7   </descriptorBase>
8 </head>
9 <body>
10  <media id="videoIntroducao" src="../videos/videoIntroducao.mp4" >
11    <area id="area01" begin="5s" end="10s" />
12    <area id="area01" begin="10s" end="14s" />
13  </media>
14 </body>
```

Figura 2: Elementos de apresentação em NCL.

Uma parte importante na construção de aplicações multimídia no NCM são os relacionamentos de causalidade entre os componentes, relacionamentos como “quando vídeo X acabar, tocar vídeo Y” ou “no meio da música A, mostrar imagem B”, definindo o sincronismo e interatividade entre os objetos da aplicação.

Esses relacionamentos são expressados por links e conectores em NCL. Conectores expressam a semântica da relação, podendo conter condições e ações, em que estas últimas ocorrem quando as condições das primeiras são concluídas, e parâmetros cujo valores devem ser definidos pelos elos, enquanto os elos associam os objetos de tal relação.

A Figura 3 apresenta a comparação da declaração de um elo em sNCL assim como seu relativo em NCL na figura 5. Nota-se em sNCL a remoção do conector e do elo, sendo os dois representados por um só elemento. Em NCL o primeiro serve para modelar o relacionamento causal da relação, enquanto o segundo serve para identificar quais elementos pertencem à relação.

Em sNCL o novo elemento representa ambos ao mesmo tempo, tanto o modelo do relacionamento quanto quais elementos pertencem à ele.

Relacionamentos simples são relacionamentos que apresentam apenas uma condição e uma ação. A figura 4 apresenta um trecho de código em NCL demonstrando o uso de conectores e elos simples. Nela é definido uma base de conectores com apenas um conector, que tem somente uma condição e uma ação.

No corpo é definido o elo que usa esse conector e explicita quais mídias fazem parte desse relacionamento. Esse elo, junto com conector, definem que quando a mídia *videoPrincipal* acabar, a mídia *imgInteratividade* também irá parar de aparecer.

```
1 onEnd videoPrincipal do
2   stop imgInteratividade
3 end
```

Figura 3: Exemplo de um elo simples em sNCL.

```
1 <head>
2   <connectorBase>
3     <causalConnector id="onEndStop">
4       <simpleCondition role="onEnd" />
5       <simpleAction role="stop" />
6     </causalConnector>
7   </connectorBase>
8 </head>
9 <body>
10  <link id="ExampleLink" xconnector="onEndStop">
11    <bind role="onEnd" component="videoPrincipal" />
12    <bind role="stop" component="imgInteratividade" />
13  </link>
14 </body>
```

Figura 4: Exemplo de um elo simples em NCL.

Em NCL, há também a possibilidade de encadear condições e ações. Condições podem ser encadeadas usando os operadores *and* e *or*, que se assemelham aos operadores lógicos com o mesmo nome. O primeiro necessita que todas as condições sejam satisfeitas para iniciar as ações, enquanto o segundo necessita somente que uma condição seja satisfeita para que as ações iniciem.

Já para as ações, a ordem de execução delas podem ser definidas de dois jeitos: paralelamente e sequencialmente. Paralelamente, todas as ações acontecem ao mesmo tempo, já sequencialmente elas acontecem uma após a outra. O que define a ordem de execução das ações é a propriedade *operator*.

```
1 onEnd videoPrincipal and onStart videoSecundario do
2   start imgMenu and start imgSubMenu
3   stop imgInteratividade
4 end
```

Figura 5: Exemplo de um elo complexo em sNCL

```

1 <head>
2   <connectorBase>
3     <causalConnector id="onBeginStartNStop">
4       <simpleCondition role="onBegin" />
5       <compoundAction operator="par" >
6         <simpleAction role="start" max="unbounded" qualifier="par"/>
7         <simpleAction role="stop" max="unbounded" qualifier="par"/>
8       </compoundAction>
9     </causalConnector>
10  </connectorBase>
11 </head>
12 <body>
13 <link id="startMomentoGol" xconnector="onBeginStartNStop">
14   <bind role="onBegin" component="videoPrincipal" interface="
15     momentoGol"/>
16   <bind role="start" component="musicaGol" />
17   <bind role="start" component="fotoGol" />
18   <bind role="stop" component="musicaPrincipal" />
19 </link>
20 </body>

```

Figura 6: Exemplo de um elo complexo em NCL.

3.2 Macros

Além dos elementos nativos da NCL, houve também a introdução de um novo elemento à linguagem chamado de macro, introduzido para solucionar os problemas de reuso causado pela remoção de alguns dos elementos.

Macros se assemelham às funções em linguagens de programação imperativas, porém como não há execução do código, ocorre somente substituição do texto em tempo de compilação. A figura a seguir mostra um exemplo de duas macro em sNCL, chamadas de *makeBasicProperties*, que recebe dois argumentos, *makeMedia*, que recebe três argumentos. A última é chamada na linha 11, onde, no processo de compilação, é processada toda a cadeia de chamadas, até chegar em um resultado sem mais chamadas, daí essa saída é inserida no lugar da chamada original.

Um dos casos de uso de macros é no caso de bibliotecas de componentes padrões que podem ser reutilizados, por exemplo dentro de emissoras, criando assim uma base comum de macros e importando essas bases em novos projetos, assim acelerando o processo de autoria das aplicações.

```

1 macro makeBasicProperties(left, right)
2   left: left
3   right: right
4 end
5 macro makeMedia(id, left, right, source)
6   media id
7   src: source
8   makeBasicProperties(left, right)
9   end
10 end
11 makeMedia("media1", 20%, 20%, "medias/imagel.jpg")

```

4 | CONSIDERAÇÕES FINAIS

Considerando os trabalhos relacionados, a SNCL diferencia-se pela proposta de uma DSL que seja mais familiar para programadores que preferem uma abordagem imperativa. Embora o Lua2NCL use a linguagem Lua, ele limita-se ao uso das tabelas para representar os elementos de NCL, criando um certo nível de complexidade na compreensão dos códigos. A SNCL propõe uma maior simplicidade na descrição das aplicações, utilizando uma linguagem de mais alto nível que não se reduza a verbosidade mas que seja também de fácil entendimento aos desenvolvedores.

A SNCL manterá as funcionalidades do NCL 3.0, como a possibilidade de reuso e importação de bases que ficam em outros documentos SNCL.

Levando em conta que os códigos de aplicações em sNCL serão transcritos para NCL, este trabalho não apresenta a sNCL como uma substituta da NCL, que já possui um player próprio, o Ginga-NCL, mas sim como uma alternativa para desenvolvedores mais familiarizados com linguagens imperativas.

REFERÊNCIAS

B. Georgea and L. Williamsb. **A structured experiment of test-driven development.**

R. Ierusalimsky. **A text pattern-matching tool based on parsing expression grammars.**

ABNT. 15606-2, 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.** 2011.

G. A. F. Lima. **Eliminando redundâncias no perfil ncl edtv.** 2011.

L. F. G. Soares, C. de Salles Soares Neto, and M. Moreno. **Ginga-ncl: Declarative middleware for multimedia iptv services.**

E. C. O. Silva, J. A. F. dos Santos, and D. C. Muchaluat-Saade. **Jns: An alternative authoring language for specifying ncl multimedia documents.** 2013.

B. Ford. **Parsing expression grammars: A recognition-based syntactic foundation**. 2004.

Lpeg: Parsing expression grammars for lua. Available at <http://www.inf.puc-rio.br/~roberto/lpeg>.

D. H. D. Bezerra, D. M. T. de Sousa, G. L. de S. Filho, A. M. F. Burlamaqui, and I. R. de M. Silva. **Luar: A language for agile development of ncl templates and documents**.

D. de Sousa Moraes, A. Damasceno, A. Busson, and C. S. Neto. **Lua2ncl: Framework for textual authoring of ncl applications using lua**. 2016.

R. Ierusalimschy, L. H. de Figueiredo, , and W. Celes. **Lua: an extensible extension language**.

F. Sant'Anna, R. Cerqueira, and L. F. G. Soares. **Nlua: objetos imperativos Lua na linguagem declarativa NCL**. 2008.

L. F. G. Soares and R. F. Rodrigues. **Nested context model 3.0: Part 1—ncm core**. Monografias em Ciência da Computação do Departamento de Informática, PUC-Rio, (18/05), 2005.

L. F. G. Soares and R. F. Rodrigues. **Nested context language 3.0 part 8—ncl digital tv profiles**. Monografias em Ciência da Computação do Departamento de Informática da PUC-Rio, page 06, 2006.

ITU-T. H. 761, **Nested context language (ncl) and ginga-ncl for iptv services**. Geneva, apr. 2009, 2009.

D. C. Bulterman and L. Hardman. **Structured multimedia authoring**. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 1(1):89–109, 2005.

K. Beck. **Test Driven Development: By Example**. 2003.

C. de Salles Soares Neto, H. F. Pinto, and L. F. G. Soares. **Tal processor for hypermedia applications**.

ÍNDICE REMISSIVO

A

- Acai berry* 74
- Accessibility* 2, 32, 140
- Adaptability* 112
- Adhesive joints* 126, 136, 138, 139
- Advertisement videos* 96
- Animals* 2
- Aquaculture reproduction* 48
- Arduino* 2, 4, 5, 12, 47, 49, 52, 57, 61, 74, 77, 80, 82
- Autistic spectrum disorder* 32, 140
- Automated monitoring* 47, 48
- Automation* 74, 191
- Automation software* 191

C

- Clustering* 14, 15, 29, 30, 31
- Cognition* 111, 112
- Cohesive zone models* 126, 138, 139
- Compilers* 84
- Cyber-crime* 169

D

- Data science* 15
- Digital image correlation* 126, 128, 130
- Digital TV* 84, 94

E

- Emotional branding* 95, 96, 99, 101, 102, 108
- Employers* 116

F

- Feature extraction* 15
- Final project report* 191
- Finite element method* 126, 127

G

Geovisualization 111, 112

Gestión de riesgos 63, 65, 68, 69, 70, 71

Gestión proyecto 152

Graduates 116

I

Informática 11, 30, 46, 63, 65, 77, 82, 94, 152, 169, 170, 171, 172, 187, 189

Information technologies 191

Innovation 74, 110

Interface 4, 32, 33, 35, 36, 38, 40, 45, 52, 76, 112, 114, 115, 128, 138, 140, 141, 143, 144, 145, 146, 149, 150, 175, 177, 178, 180, 185, 186

M

Machine learning technique 47, 48

Máquinas de guerra 209, 214, 215

Migración sistema legado 152

N

Narrativas acadêmicas 209

Neuromarketing 95, 96, 98, 99, 101, 102, 107, 108, 109, 110

P

Panvel Pharmacy 96

PEG 84, 89

Prototype 2, 74, 140

R

Retail 63, 64, 65, 69, 71

Rootkit 169, 170, 180, 184, 185, 186, 188

S

Scouts 74

Seguridad informática 63, 65

Sistema bedelías 152

Sistema de gestión de la enseñanza 152

Sistema misión crítica 152

Structural adhesives 126, 127, 128

U

Usability assessment 32

V

Virtual learning space 191

 www.atenaeditora.com.br
 contato@atenaeditora.com.br
 [@atenaeditora](https://www.instagram.com/atenaeditora)
 www.facebook.com/atenaeditora.com.br

Collection:

APPLIED COMPUTER ENGINEERING

 www.atenaeditora.com.br

 contato@atenaeditora.com.br

 [@atenaeditora](https://www.instagram.com/atenaeditora)

 www.facebook.com/atenaeditora.com.br

Collection:

APPLIED COMPUTER ENGINEERING