

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

Pedro Henrique Ventura Rodrigues de Almeida

**Modelo de representação e orquestrador baseado em nuvem para a
contação pervasiva de histórias**

Juiz de Fora

2021

Pedro Henrique Ventura Rodrigues de Almeida

**Modelo de representação e orquestrador baseado em nuvem para a
contação pervasiva de histórias**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação. Área de concentração: Sistemas Multimídia e Hipermídia

Orientador: Dr. Marcelo Ferreira Moreno

Coorientador: Dr. Carlos Pernisa Júnior

Juiz de Fora

2021

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF com os dados fornecidos pelo(a) autor(a)

Almeida, Pedro Henrique Ventura Rodrigues de.

Modelo de representação e orquestrador baseado em nuvem para a contação pervasiva de histórias / Pedro Henrique Ventura Rodrigues de Almeida. – 2021.

111 f. : il.

Orientador: Marcelo Ferreira Moreno

Coorientador: Carlos Pernisa Júnior

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2021.

1. pervasive storytelling. 2. modeling. 3. cloud-based orchestrator. 4. IoT I. Moreno, Marcelo Ferreira, orient. II. Doutor

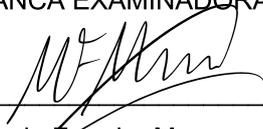
Pedro Henrique Ventura de Almeida

“Modelo de representação e orquestrador baseado em nuvem para a contação pervasiva de histórias”

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Ciência da Computação.

Aprovada em 17 de novembro de 2021.

BANCA EXAMINADORA



Prof. Dr. Marcelo Ferreira Moreno – Orientador
Universidade Federal de Juiz de Fora



Prof. Dr. Carlos Pernisa Júnior - Coorientador
Universidade Federal de Juiz de Fora



Prof. Dr. Eduardo Barrére
Universidade Federal de Juiz de Fora



Profª. Dra. Débora Christina Muchaluat Saade
Universidade Federal Fluminense

Agradeço a FAPEMIG pelo apoio. A minha família,
amigos e Deus.

AGRADECIMENTOS

Agradeço aos meus familiares pelo apoio incondicional, mesmo sem nunca entenderem o que eu faço. Obrigado em especial a minha mãe Juliana, padrasto Marcelo, avó Lurdinha e avô Altino, por sempre cuidarem de mim e me mostrarem o valor do estudo e trabalho duro. Agradeço a minha namorada Luana, pela paciência e suporte nos momentos difíceis e noites sem Netflix, “Hoje não meu bem, tenho que terminar o que o Moreno pediu”.

Agradeço a Deus por me guiar, e nunca soltar a minha mão em cada passo dado em minha vida.

Agradeço aos professores do Programa de Pós-Graduação em Ciência da Computação pelos ensinamentos transformadores. Em especial ao meu orientador, Marcelo Moreno, pela enorme paciência, dedicação e confiar em mim mais do que eu mesmo confiava, me mostrando o quanto eu era capaz.

Agradeço aos membros do LMD/FaCom pelo apoio, suporte e paciência para compreender as loucuras da computação e minhas confusões.

Agradeço a todos os colegas do Mestrado em Ciência da Computação, pela parceria, ajuda e conversas tomando um cafezinho. Em especial a nossa panelinha, Fred Sales, Mayara Amanda e Airton Ribeiro pelo apoio e contribuição nessa jornada.

Agradeço também aos meus companheiros do "Nosso" canal da Twitch, que estiveram sempre comigo nos momentos de estresse, diversão, e loucuras. Em especial aos meus amigos, Maria Clara “thelastmoria”, Gustavo “HakkoFrost”, Nikolas “nikonikonikollas”, Vanessa “lily024_”, Fabrício “someone1401” e Pureza “thepureza”.

Muitas coisas não ousamos empreender por parecerem difíceis; entretanto, são difíceis porque não ousamos empreendê-las. (SENECA).

RESUMO

Dispositivos de Internet das Coisas (IoT) estão cada vez mais acessíveis e fazem parte do dia a dia das pessoas. Isso abre grandes possibilidades para experiências inovadoras de contação de histórias, permitindo novas formas de consumo, com novos tipos de eventos e efeitos, indo além da multimídia convencional. Tratam-se de narrações capazes de se adaptarem ao contexto onde se encontra o indivíduo que a consome (este denominado interator), variando conforme as próprias escolhas. Assim, cada interator possuiria histórico próprio, resultado de suas decisões, dados informados, ou experiências com outros interatores, permitindo contações personalizadas de uma história. Nesse contexto, é perceptível a necessidade de avanços para a representação e orquestração da contação pervasiva de histórias, que considerem a singularidade de cada indivíduo e sua interação com o meio físico e o meio virtual. Este trabalho propõe um modelo conceitual denominado PST, que visa representar histórias de forma a serem contadas de maneira pervasiva, direcionado a facilitar a criação de ferramentas para autores e provedores de conteúdo. Ao modelar o domínio específico da contação pervasiva de histórias em um alto nível de abstração, o modelo apresenta suporte às variabilidades típicas de ambientes pervasivos, como mudanças na localização, na conectividade de dispositivos, na proximidade entre interatores, entre outras. O modelo é dividido em dois núcleos principais, a Aura, com entidades responsáveis por armazenar todos os dados e histórico do interator para uso na compreensão de contexto e personalização da contação de histórias; e entidades do Percorso Narrativo, responsável por estruturar a contação e guiar sua apresentação. Devido ao alto nível de abstração do modelo PST, o trabalho propõe, como prova de conceito, uma máquina de apresentação em nuvem, capaz de interpretar e orquestrar as contações de histórias representadas por meio do modelo, com experimentos de execução em nuvem e borda de histórias desenvolvidas pelo Laboratório de Mídia Digital (LMD) da Faculdade de Comunicação da UFJF. O trabalho aponta também contribuições obtidas em *feedbacks* por profissionais da área de comunicação, ainda que restritas devido à pandemia de Covid-19 e à impossibilidade de realização plena de grupos focais.

Palavras-chave: contação pervasiva de histórias, modelagem, orquestrador em nuvem, IoT.

ABSTRACT

Internet of Things (IoT) devices are becoming more affordable and part of people's daily lives. This opens up great possibilities for innovative storytelling experiences, allowing for new forms of consumption, with new types of events and effects, going beyond conventional multimedia. Such narratives would be capable to adapt to the context in which the individual consumer (called interactor) is found, varying according to their own choices. Thus, each interactor would have their own history, as a result of their decisions, informed data, or experiences with other interactors, allowing for a personalized storytelling. In this context, the need for advances in the representation and orchestration of pervasive storytelling is noticeable, taking into account the uniqueness of each individual and their interaction with the physical and virtual environments. This work proposes a conceptual model called PST, which aims to represent stories to be told in a pervasive way, aimed at facilitating the creation of tools for authors and content providers. By modeling the specific domain of pervasive storytelling as a high-level abstractions, the model supports the typical variability of pervasive environments, such as changes in location, device connectivity, proximity between interactors, among others. The model is divided into two main groups: the Aura, with entities responsible for storing all the interactor's data and history to be used for context awareness and storytelling personalization; and entities of the Narrative Course, responsible for structuring the narration and guiding its presentation. Due to PST's high-level abstraction, the work proposes, as a proof of concept, a cloud-based presentation machine, capable of interpreting and orchestrating the storytelling as represented through the model, with cloud and edge execution experiments of stories developed by the Digital Media Laboratory (LMD) of the Faculty of Communication at UFJF. The work also points out contributions obtained in *feedbacks* by professionals in the field of communication, despite the restrictions imposed by the Covid-19 pandemic and the impossibility of fully conducting focus groups.

Keywords: pervasive storytelling, modeling, cloud-based orchestrator, IoT.

LISTA DE ILUSTRAÇÕES

Figura 1	– Estrutura de um Orquestrador em Nuvem	24
Figura 2	– Contação de história simulada em um cenário doméstico - experiência 1	25
Figura 3	– Contação de história simulada em um cenário de transporte público - experiência 2	26
Figura 4	– Contação de história simulada em um cenário de pausa do trabalho	27
Figura 5	– Contação de história simulada em um cenário de ambiente externo	28
Figura 6	– Estrutura do corpo (<i>body</i>) de um documento HTML5	30
Figura 7	– Visão geral da hierarquia de classes do NCM	32
Figura 8	– Estrutura básica de um documento NCL	34
Figura 9	– Máquina de estados de eventos NCL	35
Figura 10	– Elementos NCL relacionados às extensões NCM propostas por Guedes (19).	37
Figura 11	– Exemplo de aplicação de mídias DOOH em ambiente urbano. . .	41
Figura 12	– Fragmento do Premiere Pro de uma cena composta por trilhas com seus respectivos objetos de mídias.	42
Figura 13	– Entidades do Modelo STorM.	42
Figura 14	– Diagrama de elementos da linguagem STorML.	44
Figura 15	– Máquina de estados de STorML.	45
Figura 16	– Visão geral da arquitetura do MPEG-V.	47
Figura 17	– Representação do Conceito da SEDL do MPEG-V.	49
Figura 18	– Luta por Detroit - Kara Deixa Detroit, Detroit Become Human . .	59
Figura 19	– Entidades estruturais do Modelo PST agrupadas em visão composicional	60
Figura 20	– Principais Entidades Estruturais do Modelo PST	61
Figura 21	– As Entidades do Interactor e Aura	70
Figura 22	– Máquina de Estados da Apresentação	72
Figura 23	– Fragmento da Primeira Narrativa criada Pelo LMD - Pt. 1	77
Figura 24	– Fragmento da Primeira Narrativa criada Pelo LMD - Pt. 2	78
Figura 25	– Primeira Narrativa criada Pelo LMD, Lux Ferre - Completa	79
Figura 26	– Arquitetura da Máquina de Apresentação Baseada em Nuvem . .	83
Figura 27	– Dispositivos de saída utilizados na pesquisa.	84
Figura 28	– Console da Máquina de Apresentação.	85
Figura 29	– Tela do aplicativo Mobile de captura de entrada	86
Figura 30	– Bancada de teste de atraso de execução dos dispositivos	87
Figura 31	– Resultados do teste de atraso de execução dos dispositivos - Local	88
Figura 32	– Resultados do teste de atraso de execução dos dispositivos - Nuvem	89
Figura 33	– Avaliação de Compreensão do Modelo Pervasivo - Parte 1	105
Figura 34	– Avaliação de Compreensão do Modelo Pervasivo - Parte 2	106
Figura 35	– Resposta da Avaliação de Compreensão do Modelo Pervasivo - Parte 1	106
Figura 36	– Resposta da Avaliação de Compreensão do Modelo Pervasivo - Parte 2	107
Figura 37	– Diagrama de classes do Modelo PST	108

Figura 38	- Representação parcial da narrativa de Anna - Parte 1	109
Figura 39	- Representação parcial da narrativa de Anna - Parte 2	111

LISTA DE TABELAS

Tabela 1 – Relações Temporais do Modelo SIMM	53
Tabela 2 – Alinhamento do Modelo SIMM	53
Tabela 3 – Resumo comparativo dos Trabalhos Relacionados no âmbito de contação pervasiva de histórias	56
Tabela 4 – Estados e Transições da Máquina de Estados.	75
Tabela 5 – Entidades, Hierarquia e Atributos do Modelo PST	76

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	Application Programming Interface
CIDL	Control Information Description Language
CSS3	Cascading Style Sheets 3
DOOH	Digital Out-Of-Home
DCDV	Device Capability Description Vocabulary
DCV	Device Command Vocabulary
HTML	Hypertext Markup Language
HTML5	Hypertext Markup Language version 5
HTTP	Hypertext Transfer Protocol
IIDL	Interaction Interface Description Language
IPTV	Internet Protocol Television
IoT	Internet of Things (Internet das Coisas)
JSON	JavaScript Object Notation
LED	Light-emitting diode (Diodo emissor de luz)
LMD	Laboratório de Mídia Digital (UFJF)
NCL	Nested Context Language
NCM	Nested Context Model
REST	Representational State Transfer
RGB	Red, Green, Blue.
SBTVD	Sistema Brasileiro de Televisão Digital
SCDV	Sensor Capability Description Vocabulary
SEDL	Sensory Effect Description Language
SEPV	Sensory Effect Preference Vocabulary
SEV	Sensory Effect Vocabulary
SIV	Sensed Information Vocabulary
SMIL	Synchronized Multimedia Integration Language
STorM	Scene- and Track-oriented hypermedia model
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
YAML	YAML Ain't Markup Language

SUMÁRIO

1	INTRODUÇÃO	14
1.1	PROPOSTA DA DISSERTAÇÃO	15
1.2	ORGANIZAÇÃO DA DISSERTAÇÃO	15
2	FUNDAMENTAÇÃO TÉCNICO-TEÓRICA	16
2.1	INTERNET DAS COISAS	16
2.1.1	Sensores	16
2.1.2	Atuadores e Efeitos Sensoriais	17
2.1.3	Arquitetura de aplicações IoT	17
2.1.3.1	HTTP/REST	19
2.2	O AMBIENTE PERVASIVO ENCONTRA NARRATIVAS E JOGOS DIGITAIS	20
2.3	CENÁRIOS DE USO DE UM MODELO DE REPRESENTAÇÃO DA CONTAÇÃO PERVASIVA DE HISTÓRIAS	22
2.3.1	Requisitos para um Modelo de Representação da Contação Pervasiva de Histórias	28
3	TRABALHOS RELACIONADOS	30
3.1	HTML5	30
3.2	NCM - NESTED CONTEXT MODEL (3.0)	31
3.3	NCL - NESTED CONTEXT LANGUAGE	33
3.3.1	NCL Estendida para Suporte a Multiusuários e Interações multimodais	34
3.3.2	NCL Estendido para Suporte a Efeitos Sensoriais	37
3.4	SMIL	39
3.5	MODELO STorM	40
3.6	LINGUAGEM STorML	43
3.7	MPEG-V	46
3.8	MODELO SIMM	52
3.9	MULTISEM	52
3.10	COMPARATIVO DOS TRABALHOS RELACIONADOS NO ÂMBITO DE CONTAÇÃO PERVASIVA DE HISTÓRIAS	54
4	MODELO PERVASIVE STORYTELLING - PST	57
4.1	PERCURSO NARRATIVO	58
4.2	ENTIDADES ESTRUTURAIS	59
4.2.1	Pervasive Storytelling	61
4.2.2	Episode	62
4.2.3	Experience	62
4.2.4	Sequence	64
4.2.5	Circunstâncias	64
4.2.6	Entidades de Entrada e Saída	65
4.2.6.1	<i>Mídias e Efeitos Sensoriais</i>	66
4.2.6.2	<i>Ponto de Decisão</i>	67
4.3	INTERACTOR E AURA	69
4.3.1	Múltiplos Interatores	71
4.4	MÁQUINA DE ESTADOS DAS ENTIDADES	71

4.5	Resumo de Entidades, Hierarquia e Atributos de PST	76
5	REPRESENTANDO UMA CONTAÇÃO DE HISTÓRIA COM O MO- DELO PST	77
5.1	PESQUISA EXPLORATÓRIA SOBRE <i>FEEDBACKS</i> E COMPREENSÃO DO MODELO PST	79
6	Orquestrador Baseado em Nuvem	82
6.1	INTERACTOR'S AURA LAYER	83
6.2	DEVICE CONTROL ABSTRACTION LAYER	83
6.2.1	Latência dos Dispositivos	87
6.3	MÁQUINA DA APRESENTAÇÃO	89
6.4	EXEMPLOS DE EXECUÇÃO	91
7	CONCLUSÃO	93
7.1	Contribuições da Dissertação.	94
7.2	Trabalhos Futuros	95
	REFERÊNCIAS	97
A	Fragmento de Lux Ferre	102
B	Formulário de Avaliação da Compreensão do Modelo Pervasivo e Resultados	105
C	Diagrama de classes do Modelo PervasiveStorytelling	108
D	Representação da narrativa de Anna seguindo o Modelo Pervasi- veStorytelling	109

1 INTRODUÇÃO

Atualmente, dispositivos de Internet das Coisas (IoT) estão cada vez mais acessíveis e fazendo parte do dia a dia das pessoas. Relógios e lâmpadas “inteligentes”, celulares, TVs, controles de infravermelho, são apenas alguns dos muitos dispositivos no mercado, sendo comumente integrados através de um assistente pessoal implementado através de computação em nuvem.

Tal avanço tecnológico abre grandes possibilidades para experiências inovadoras, permitindo novas formas de contar histórias, com novos eventos e efeitos, indo além da multimídia convencional (áudio/vídeo/texto/imagem). Em aplicações multimídia convencionais, normalmente um dispositivo capaz de reproduzir sons e imagens orquestra e apresenta o conteúdo localmente, ou mesmo integrando dispositivos secundários (54).

Contações interativas de histórias com uso destes dispositivos vêm sendo exploradas recentemente, como as *Skills* interativas Gumball Medieval (60) e Trivia Ursos Sem Curso (61), nas quais uma narração oral é apresentada, com as vozes dos personagens de animações originais e chamadas à interação dos ouvintes. Apesar da limitação de formas de interação, exclusivamente por voz, e das dificuldades inerentes ao reconhecimento da fala, tais obras são um vislumbre das capacidades de consumo de tais conteúdos com dispositivos IoT.

Neste contexto de uso intensivo da computação em nuvem e de conectividade dos dispositivos IoT, não é difícil notar a necessidade de avanços para a representação e orquestração que viabilizem uma contação pervasiva de histórias. Segundo Pløhn et al. (44), uma contação pervasiva de histórias, no contexto de seu estudo em narrativas para jogos, necessita fazer parte do mundo real em que realmente vive o jogador, criando uma imersão narrativa de uma história onipresente, indo além da consciência no jogo, onde as tecnologias, assim como descrito por (38), tornam-se transparentes ao jogador/espectador.

Atualmente, mesmo narrativas interativas simples com uso básico de recursos dos assistentes de voz, demandam considerável esforço de implementação, precisando ser desenvolvidas por programadores especializados. Por outro lado, soluções de autoria mais elaboradas e em alto nível, que permitem certo grau de personalização e pervasividade da contação de uma história, possuem limitações quanto a dispositivos específicos de reprodução, máquinas de apresentação confinadas em dispositivos de usuário e domínio específico em hiper/multimídia, não exatamente focadas na representação e nas variabilidades da contação pervasiva de histórias.

De fato, dada a liberdade de mobilidade dos usuários e a gama de dispositivos que podem ser agregados não somente pelo próprio usuário, mas também pelo provedor da contação de histórias em espaços dispersos, nota-se a importância de que a orquestração da apresentação seja realizada de forma ubíqua. No atual estágio de conectividade de dispositivos, a concepção de uma máquina de apresentação em nuvem é uma forma viável de se estabelecer tal ubiquidade, ressalvadas as variações inerentes ao melhor esforço da Internet, à comunicação sem fio, disponibilidade de energia, entre outros.

1.1 PROPOSTA DA DISSERTAÇÃO

Para contribuir no contexto apresentado na introdução 1, este trabalho propõe um modelo conceitual especificamente voltado à representação da contação pervasiva de histórias, com utilização de dispositivos conectados em rede e suporte a múltiplos usuários. O modelo, denominado PST (*Pervasive StoryTelling*), visa facilitar a concepção de ferramentas para a criação pelo autor, uma vez que uma desejada linguagem visual não está no escopo desta dissertação especificamente. O modelo foi criado a partir da observação do processo criativo de pesquisadores em Comunicação do Laboratório de Mídia Digital da Universidade Federal de Juiz de Fora, de modo que a concepção e, posteriormente, a validação das estruturas e terminologia adotadas tivessem a participação de especialistas no domínio de narrativas.

Ao modelar o domínio específico da contação pervasiva de histórias em alto nível de abstração, o modelo introduz suporte às variabilidades típicas de ambientes pervasivos, como as mudanças de localização, a conectividade de dispositivos, a proximidade entre usuários, entre outras.

Além disso, este trabalho também propõe uma máquina de apresentação em nuvem, capaz de interpretar as contações de histórias representadas por meio do modelo PST, para orquestrar o consumo de forma desacoplada de dispositivos específicos, controlando-os conforme disponibilidade e definições dos autores.

1.2 ORGANIZAÇÃO DA DISSERTAÇÃO

A dissertação está organizada da seguinte forma: No Capítulo 2 é fornecida uma breve conceituação de tecnologias empregadas no desenvolvimento deste trabalho e uma contextualização do ambiente ao qual o trabalho foi idealizado. O Capítulo 3 discute algumas linguagens de autoria de conteúdo para o domínio de multimídia, hipermídia, uso de efeitos sensoriais, e também trabalhos de outros domínios que usam alguns conceitos fundamentais ou semelhantes aos definidos nesta dissertação. O Capítulo 4 trata a definição do modelo conceitual *Pervasive Storytelling* (PST). No Capítulo 5 é apresentada uma contação de história desenvolvida utilizando o modelo PST por profissionais da comunicação, e *feedbacks* dos autores referentes ao modelo. No Capítulo 6 é apresentado um orquestrador em nuvem desenvolvido segundo o modelo PST proposto no Capítulo 4, para reprodução das contações. Finalmente, o Capítulo 7 é dedicado às considerações finais do estudo e indicações de trabalhos futuros.

2 FUNDAMENTAÇÃO TÉCNICO-TEÓRICA

Neste tópico serão introduzidos conceitos e visões tomados como base para a concepção deste trabalho, que vão desde as novas possibilidades com a Internet das Coisas (IoT) até as dinâmicas abordadas em jogos e contações pervasivas de histórias em si. O capítulo é finalizado com uma breve apresentação de cenários de uso para um modelo de representação voltado à contação pervasiva de histórias, de forma a proporcionar ao leitor uma melhor compreensão da pesquisa e desenvolvimento do raciocínio empregado neste trabalho.

2.1 INTERNET DAS COISAS

Como definido por Santaella et al. (52) a Internet das Coisas (IoT) representa uma extensão da Internet como a conhecemos, capaz de proporcionar aos objetos do dia-a-dia com capacidade computacional e de comunicação, se conectarem à rede Internet. Com isso, torna-se possível controlar remotamente os objetos e que estes sejam acessados como provedores de serviços. Tais objetos são divididos em dois grandes grupos, sensores (entrada) e atuadores (saída).

Tais dispositivos vêm cada vez mais se popularizando, com diferentes casos de uso, que vão além do âmbito pessoal, como automação industrial, cidades inteligentes (*Smart Cities*), saúde (*Healthcare*) e casas inteligentes (*Smart Home*). No entanto, ainda há desafios de regulamentações, segurança e de padronizações (46).

2.1.1 Sensores

Uma considerável parte dos estudos em computação visa a representação da informação do mundo real. Várias dessas representações são obtidas a partir de sensores baseados em como o ser humano capta as informações do mundo, os sentidos humanos. Sensores de imagem em câmeras fotográficas e sensores de áudio em microfones são exemplos consolidados no dia-a-dia das pessoas com seus dispositivos.

Seguindo a visão aristotélica, o homem possui 5 sentidos: visão, audição, tato, paladar e olfato. Contudo, segundo Durie (16), a quantidade de sentidos se estende além da definição anterior, sendo os sentidos baseados na capacidade de captação e interpretação dos órgãos humanos sobre eventos que ocorrem no mundo ao seu entorno. As informações são agrupadas em três meios: químico (gosto, cheiros ou “internamente”), mecânico (toque ou audição) e luz (visão). Para Durie (16), tais sensações e percepções podem ser obtidas por vinte e um sentidos do corpo humano, todos relacionados a qual tipo de informação o corpo humano consegue perceber e compreender, como: espectros de luz, som, as variações da percepção do paladar (como salgado, doce, amargo), variações do tato como pressão, variações da dor, gravidade, percepção de aceleração, temperatura, sensações corporais internas, entre outros.

Logo, Sensores, que podem ser generalizados como *dispositivos de entrada*, tratam-se de dispositivos capazes de captar tais informações tal qual, ou melhor, que os sentidos humanos, de modo a permitir sua representação e armazenamento

digitalmente. Alguns exemplos de sensores comuns capazes de monitorar o ambiente em que se encontram são sensores de temperatura, pressão, umidade, luz, entre outros. Além disso, alguns dispositivos de entrada podem tratar da captação de ações e intenções do corpo humano visando a manipulação de ambientes virtuais, tais como um sensor de imagem para captação de gestos, os sensores de movimentos de um *joystick*, teclado ou mesmo *mouse*.

2.1.2 Atuadores e Efeitos Sensoriais

Os Atuadores, que poderiam ser generalizados como *dispositivos de saída*, possuem o trabalho de traduzir dados digitais em ações e fenômenos a serem percebidos pelo homem no mundo real. Atualmente, os atuadores capazes de reproduzir o que se pode chamar de mídias convencionais são os mais difundidos entre consumidores comuns, tratando-se basicamente da geração de estímulos audiovisuais (texto, imagem, vídeo e som), sendo eles TVs, celulares, monitores, caixas de som, entre muitos outros.

Além das mídias convencionais, há também atuadores capazes de reproduzir “Efeitos Sensoriais” (15). Tratam-se de efeitos que geram percepção através de outros estímulos aos sentidos humanos, como vento, olfato, paladar, luz, tato, calor, entre outros. No atual estado da arte, devido à falta de padrões e evolução em usabilidade, grande parte dos atuadores de efeitos sensoriais necessita de alguma intervenção por pessoas especializadas, capazes de compreender os mecanismos individualmente, para que seja possível sua exploração em apresentações imersivas e com os efeitos desejados (34).

Contudo, conforme descrito em ISO/IEC 23005-1 (22), um criador de um conteúdo não pode se limitar a um único modelo de atuador. De fato, quando se produz um conteúdo audiovisual, o mesmo deve ser capaz de ser reproduzido em diversos *players* diferentes, cada um com suas limitações ou mecanismos extras, mas não há uma obrigatoriedade de se definir especificidades do dispositivo ao *player*. Logo, essa mesma facilidade é desejada na criação de conteúdos com efeitos sensoriais.

2.1.3 Arquitetura de aplicações IoT

Ordinariamente, dispositivos IoT não possuem grande poder de processamento, devido, em parte, ao fato de serem dispositivos com funcionalidades limitadas ao contexto e à função que lhe são propostos, sendo, assim, controlados por outras aplicações. São na grande maioria das vezes elementos de um algo maior, controlado por outros dispositivos e aplicações para inúmeros cenários distintos, como explicitado por Leite (30): **Smart Home** para automação residencial, gerando maior conforto e facilidade no controle de dispositivos comuns como TVs, ar-condicionado, luzes e afins; **Smart Building** em automação de edifícios de uso coletivo ou doméstico, capazes de proporcionar um melhor gerenciamento automatizado de recursos comuns como ar-condicionado, iluminação e energia; **Smart Cities** com automações para melhoria do coletivo local de uma cidade, por exemplo, otimização de tráfego através de trocas de informações entre carros, celulares e semáforos; **Smart Grid**, relacionado à eficiência da geração e distribuição de energia por empresas do setor

elétrico, capazes de obter em tempo real consumo e estado da rede elétrica dos clientes; **Healthcare** possibilitando monitoramento e acompanhamento remoto de pacientes, obtendo em tempo real informações como temperatura, pressão arterial, batimento cardíaco, entre outros; **Smart Manufacturing** para controle e automação de processos fabris industriais; e inúmeras outros cenários em ascensão, como logística, agronegócio, construção civil, gasodutos, etc.

Para ser possível construir aplicações nos cenários supracitados, somados à dificuldade existente nos inúmeros tipos e fabricantes de dispositivos, são necessários padrões e protocolos de comunicação para interoperabilidade dos dispositivos e aplicações controladoras.

Conforme mencionado por Prates (46), as principais soluções para IoT se encontram atualmente fundadas em tecnologias que incluem, de forma proeminente, a computação em nuvem. Segundo (57) a computação em nuvem é uma metáfora, onde é possível abstrair e ocultar a complexidade de uma infraestrutura real descentralizada e compartilhada, disponibilizada como serviços através de uma rede como a Internet. De forma mais palatável a leitores de áreas distintas da computação, a computação em nuvem permite que várias máquinas forneçam serviços, ferramentas e aplicações para inúmeras pessoas, que as acessem através de diversos tipos de dispositivos, com diferentes capacidades computacionais e em diferentes lugares.

Todos os recursos computacionais são disponibilizados pelo provedor do serviço de nuvem, cabendo ao usuário possuir somente um dispositivo capaz de se conectar à rede através de um navegador ou aplicação específica, diminuindo o custo em equipamento para usuários. Partindo então desse paradigma, a união entre computação em nuvem e dispositivos IoT torna-se interessante, dado que o peso do processamento não necessita mais ficar na ponta (usuário ou dispositivos finais), e os dispositivos IoT mesmo limitados em processamento e armazenamento, mas capazes de se conectar em rede, e são, assim, controlados remotamente por outras máquinas.

De fato, atualmente inúmeros serviços são disponibilizados para *Smart Home*, com serviços em nuvem para controle dos dispositivos, como os assistentes domésticos Alexa da Amazon controlada via AWS e Google Home pela Google Cloud. Os dispositivos físicos de tais assistentes, como Echo Dot, possuem hardwares simples no quesito processamento, com um processador 32 bits ARM Cortex-A8, 256MB LPDDR1 RAM e 4GB de armazenamento ¹, sendo estas especificações técnicas bem mais modestas que celulares de entrada, demonstrando que todo o processamento de fato é feito pela nuvem da Amazon (AWS-Lambda).

Outros serviços comuns em nuvem são das próprias fornecedoras de dispositivos domésticos como de lâmpadas, tomadas e controladores infravermelhos, como das marcas Tuya e Xiaomi-Yeelight. Os serviços de nuvem das próprias fabricantes dos dispositivos permitem ao usuário final controlá-los através de uma aplicação móvel, inclusive conexão e controle por serviços de nuvem de terceiros, por meio de APIs, possibilitando assim os assistentes domésticos supracitados controlem tais dispositivos.

¹ <https://www.ifixit.com/Teardown/Amazon+Echo+Teardown/33953>

Apesar dos benefícios da computação em nuvem, a distância entre os dispositivos e o servidor pode ocasionar problemas inerentes à abordagem de melhor esforço da Internet. Em um cenário onde o tempo de resposta entre ação e execução possui importância, como questões de sincronia em aplicações multimedial (*multiple sensorial media*) ou controle em tempo real de maquinário em uma fábrica. Nesses cenários, pode-se recorrer à computação em borda (*edge computing*)(53), que estende a computação em nuvem para localizações mais próximas dos dispositivos finais.

Na computação em borda, são disponibilizados micro data centers ou máquinas mais modestas no local onde se encontram os dispositivos que interajam com o serviço. Estas máquinas processam e armazenam os dados requisitados localmente, enviando para a nuvem somente o que for necessário e quando for necessário (47). Além disso, os dispositivos na borda podem receber comandos das máquinas em nuvem para que ações sejam disponibilizadas para os dispositivos finais.

Apesar de tais tecnologias serem fundamentais para uma orquestração IoT, elas por si só não garantem o controle e obtenção dos dados de um dispositivo IoT. Para isso, são necessários os protocolos de comunicação. Diversos protocolos podem ser utilizados para proporcionar tal comunicação com dispositivos IoT, contudo, o MQTT e o HTTP/REST são os dois protocolos que se sobressaem no cenário IoT conforme explicitado por DEAL(14). As próximas subseções são dedicadas à descrição de tais protocolos.

2.1.3.1 HTTP/REST

Web Service é uma solução utilizada na integração de sistemas e comunicação entre aplicações distintas. Através dela, é possível que aplicações mais recentes possam interagir com outras mais antigas, através do envio e recepção de dados pela Web, independente de plataforma, linguagem ou sistema operacional. Os dados são transmitidos por meio do protocolo HTTP e em um formato intermediário em comum, como XML, JSON, YAML, traduzido pelo receptor para sua linguagem de domínio.

Uma das principais tecnologias utilizadas em *Web Services* é o estilo arquitetural *Representational State Transfer* (REST), ou Transferência de Estado Representacional. REST especifica um padrão de interfaceamento para aplicações e serviços Web (9), criado por Roy Fielding, para definir características fundamentais para o desenvolvimento de tais aplicações. O estilo arquitetural REST permite comunicação entre aplicações na Web, por meio da definição de APIs remotas acessíveis através de uma conexão TCP/IP e mensagens HTTP de diferentes métodos.

Uma requisição REST é composta a princípio por quatro elementos: método, URL (rota), parâmetros e tipos de dados utilizados na requisição e esperados como resposta.

O método define do que se trata a requisição, ou seja, o que se deseja obter. Quando é definido o tipo "GET", é determinado que a requisição deseja apenas obtenção de informação referente à rota chamada, sendo este o padrão das requisições feitas por navegadores ao acessar um *website* por exemplo. O

método “POST” corresponde à inserção de um novo dado junto à aplicação, “PUT” para modificação, e “DELETE” para exclusão. Apesar desses quatro serem os mais conhecidos, o padrão determina outros métodos como “CONNECT”, “HEAD”, “TRACE”, “OPTIONS” e “PATCH”, cada um com semântica própria.

Uniform Resource Locator (URL), de forma mais elucidativa, é a localização (endereço do servidor e caminho) de um recurso Web, como uma página ou de uma aplicação com a qual deseja-se realizar a comunicação. Em REST é comum referenciar a URL como uma "rota", na qual é possível também informar parâmetros para que a aplicação solicitada possa processar informações. Para exemplificar, suponha que se deseja acessar um site de compras para visualizar o preço de eletrônicos. Caso o serviço da loja digital tenha um parâmetro “eletronico” definido, no lugar de ter que navegar pelo website, o usuário poderá digitar diretamente no navegador a URL + Parâmetro, para obter o resultado desejado, ficando da seguinte forma ilustrativa: “GET www.loja.com.br/eletronico”. Vale ressaltar, que em uma consulta através de um navegador, o método GET é dado como padrão, não sendo necessário pelo usuário informá-lo.

Além da rota, os parâmetros também podem ser informados em meio ao corpo da requisição HTTP. O corpo da requisição compreende os dados enviados na requisição, mas inseridos na mensagem encaminhada, e não na URL. Os tipos de dados enviados são acordados no momento da requisição entre servidor e cliente, podendo ser, por exemplo, em JSON, XML e YAML.

O estilo arquitetural REST é fortemente utilizado para obter informações de APIs, controlar dispositivos e serviços de nuvem IoT, sendo ele o padrão utilizado neste trabalho para realizar as comunicações entre aplicações, dispositivos e serviços. Seguindo o padrão de *web services*, torna-se possível que um usuário consiga interagir com múltiplos dispositivos distintos conectados em rede, controlados por uma aplicação em uma máquina distinta com acesso à rede, e por consequência a tais dispositivos.

2.2 O AMBIENTE PERVASIVO ENCONTRA NARRATIVAS E JOGOS DIGITAIS

O paradigma de computação pervasiva prevê a capacidade do usuário, a todo e qualquer momento, e lugar, acessar os mais diversos tipos de dispositivos, sendo estes móveis ou não (66), de forma que os mesmos desapareçam para o usuário (51). Veja que este desaparecimento dos dispositivos, não se trata do usuário não observar a existência das máquinas, mas sim que as mesmas sejam tão comuns ao dia-a-dia do usuário, que a interação com elas seja algo natural, tal qual a utilização de talheres é uma extensão do corpo humano, se tornando então “transparentes”, conforme definido por Murray (38).

Através de tal transparência, a computação ubíqua e pervasiva permite expandir narrativas convencionais, propiciando que o meio onde se encontra o usuário e as tecnologias disponíveis, consigam gerar maior imersão e adaptada ao contexto, onde o usuário não percebe que está em tempo real utilizando de outros dispositivos para experienciar sua narrativa, se tornando parte da realidade.

A popularização de dispositivos móveis, pessoais e de IoT vem permitindo a

exploração de aplicações voltadas à contação pervasiva de histórias. Notoriamente, aplicações de realidade aumentada para narrativas e jogos pervasivos vêm ganhando forma. Ainda que não sigam um modelo uniforme de representação de dados para a contação pervasiva de suas histórias, tais iniciativas apontam algumas possibilidades e ajudam a formar o cenário de uso vislumbrado pelo presente trabalho.

Bichard et al. (10) propõem um jogo com utilização de sensores de geolocalização (GPS), giroscópio, celulares, *walkie-talkies* com sintetizadores de voz, e um microfone direcional. O *framework* proposto conecta os dispositivos a um servidor, para troca de dados de entrada, saída e processamento. Integrado a um Sistema de Informação Geográfica, permite transformar informações locais reais, como semáforos ou paisagens, em objetos virtuais para tomada de decisões na narrativa programada. Os jogadores, alocados no banco de trás de um carro em movimento, interagem por voz com personagens e objetos virtuais, que aparecem conforme a localização e posição dos jogadores. A lógica e a coordenação dos efeitos que compõem a experiência é feita por meio da execução de *scripts*, desenvolvidos por programadores especialistas.

Trabalhos apontados por (26) recorrem à geolocalização e mineração de dados em rede para contação pervasiva de histórias. Em 2016, o uso de geolocalização em jogos se torna popular com o título “Pokemon GO” (41) para dispositivos móveis, em que, similar a (10) e ao levantamento de (26), os jogadores têm sua geolocalização rastreada e objetos criados, baseados em informações reais de cenários próximos, passíveis de interação pelos jogadores. Nota-se, naquele momento, a introdução da computação em nuvem para a execução de pelo menos parte da lógica das narrativas.

Em (12), é abordada uma arquitetura capaz de analisar dados de contexto do usuário e obter informações *online* de elementos nas proximidades. Além disso, a arquitetura inclui a mineração de informações relacionadas, através de redes sociais conectadas, de modo a produzir uma narrativa adaptada ao contexto, personalizada ao interesse do indivíduo. O modelo demonstra a capacidade de aplicações em nuvem de criar experiências únicas, através de dados públicos na Web e em redes sociais permitidas pelo usuário. Interessantemente, dado que tais informações são constantemente modificadas com as interações do usuário e de sua respectiva rede, permite-se uma constante adaptação de uma narrativa.

O kit de desenvolvimento *Alexa* (3) permite a criação das chamadas *Skills*, o que exige conhecimento de programação. Por outro lado, são também oferecidos serviços que não exigem conhecimento específico, como o *Alexa Skills Blueprints* (4). Tratam-se de arquétipos prontos de tipos de aplicativos específicos, para que não programadores possam personalizá-los. *Blueprints* específicos para contação de histórias possuem recursos limitados à inserção de textos para *Text-To-Speak*, efeitos sonoros nativos e reações.

É importante mencionar que, nesta área, há trabalhos que não se limitam ao suporte computacional para contação de histórias. O estudo (44) visa compreender e medir como os elementos devem ser utilizados em uma contação pervasiva de histórias para um melhor êxito e imersão do usuário. Muitas destas questões foram levantadas por Murray (38), antes mesmo da popularização de dispositivos capazes

de trazer a pervasividade fora de ambientes controlados.

Por sua vez, Araújo (6) propõe uma ferramenta de auxílio a criadores de conteúdo na autoria, em etapa de projeto, e concepção de conteúdo de narrativas audiovisuais interativas (não lineares), denominada ISB Designer. A ISB Designer baseia-se em técnicas de *storyboard* utilizadas nas áreas de cinema, TV e animação, acrescidas de momentos de interação e exibição em telas secundárias.

Nota-se, até este ponto, uma ausência de modelos de alto nível capazes de representar as contações pervasivas de histórias e de serem usados na orquestração da apresentação das mesmas. Idealmente, tais modelos devem ser facilmente assimilados por profissionais de criação de conteúdo, por meio de ferramentas de autoria. E, neste mesmo caminho, percebe-se que modelos e linguagens já estabelecidos no domínio específico de hiper/multimídia vêm sendo alvo de adaptações e extensões.

2.3 CENÁRIOS DE USO DE UM MODELO DE REPRESENTAÇÃO DA CONTAÇÃO PERVASIVA DE HISTÓRIAS

Desde as rádio novelas, as histórias são contadas em torno de um dispositivo primário. Os ouvintes se reúnem em torno de um aparelho capaz de reproduzir sons captados por radiofrequência, produzido por alguém a quilômetros de distância, até ser captado por um rádio capaz de reproduzir em sons as ondas eletromagnéticas transmitidas. Com a chegada da televisão, era possível reproduzir imagens além do som. Contudo, os agora telespectadores, ainda necessitavam de estar em torno de um dispositivo capaz de contar a história.

Com o passar dos anos, dispositivos extras estenderam as televisões para aumentar a experiência do usuário, como os aparelhos de VHS e DVDs, capazes de reproduzir itens gravados, caixas de som *surround* para maior imersão audível, e até TVs capazes de controlar dispositivos diversos como iluminação. Com a TV digital, aplicações interativas passaram a ser transmitidas junto à programação normal, possibilitando ao espectador um pouco mais de engajamento com o conteúdo. No Brasil, com o Sistema Brasileiro de TV Digital e seu Ginga-NCL (54), dispositivos móveis puderam ser tornar um dispositivo secundário junto à apresentação transmitida.

Com a evolução dos jogos digitais, espectadores até então passivos frente à TV, puderam se tornar protagonistas de suas próprias histórias, e através de um console de *videogame* em uma televisão, ou computador, se tornaram capazes de interagir com ambientes virtuais e decidir o rumo da narrativa que consomem.

Por mais que desde o último século tais tecnologias tenham passado por avanços, ainda é possível perceber a necessidade comum de um dispositivo primário local para orquestrar a apresentação. Os modelos atuais necessitam do espectador estar naquele ambiente, que se limita aos dispositivos ali presentes.

Os celulares inteligentes (*smartphones*), devido a sua grande capacidade de processamento, e por possuírem diversas tecnologias em sua construção, como conexão a redes sem fio, e principalmente sua portabilidade, permitiram que aplicações mais disruptivas pudessem ser desenvolvidas. Jogos e aplicações baseadas em geolocalização permitem que usuários interajam diretamente com objetos reais e suas representações digitais, com informações atualizadas obtidas pela Internet,

removendo assim o até então espectador de sua limitação física.

Há atualmente uma grande variedade de dispositivos capazes de se conectar a rede, capazes de serem controlados por outros dispositivos, além de trocarem informações. Com essa habilidade, tais dispositivos podem ser mais que uma extensão de uma tela. Tornam-se elementos de mesmo nível, capazes de gerar conteúdo e obter informações em espaços distintos e não mais fixos em uma sala ou ambiente controlado. As histórias passam a poder ser contadas com o que existe ao redor do interator, de forma pervasiva, sem que ele perceba como a contação está controlando dispositivos e/ou que os mesmos capturem informações. A história pode ser adaptada não somente com decisões e controle de quem a consome, mas também com o que existe no contexto onde o indivíduo se encontra.

Devido à capacidade ativa de consumo do conteúdo dos cenários pervasivos vislumbrados nesta seção, que eleva a experiência comumente encontrada por usuários de aplicações multimídia convencionais, este trabalho apropria-se do conceito de “interator”, de Murray (38), para denominar o usuário consumidor e, ao mesmo tempo, participante de uma contação de história.

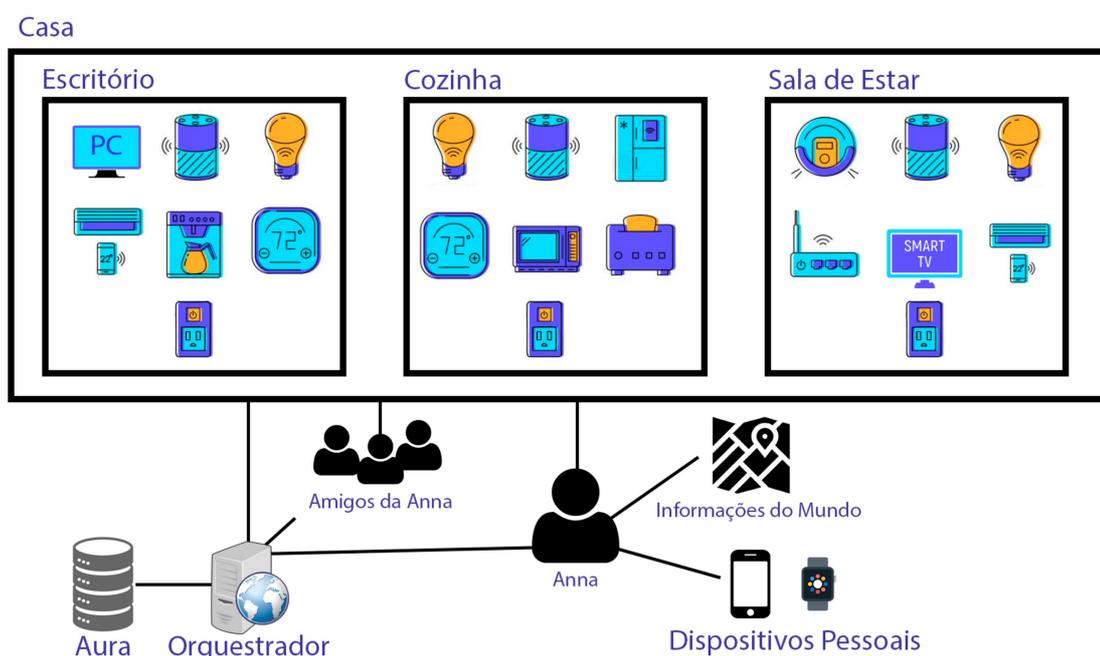
Tal formato evolui as formas convencionais de contação de história, removendo a necessidade de um objeto central que a apresente, como uma TV, computador, videogame ou rádio. No cenário apresentado, todos os dispositivos disponíveis no momento da apresentação tornam-se uma ferramenta, não havendo um dispositivo físico local responsável por orquestrar a contação. Com isso, é necessário um elemento exclusivo para orquestração, capaz de acessar qualquer dispositivo de forma ubíqua.

Com a capacidade dos dispositivos de se comunicarem pela Internet, torna-se viável um orquestrador capaz de se conectar com tais dispositivos pela rede mundial de computadores. Uma máquina capaz de os controlar independente do ambiente, desde que seja possível uma conexão em rede. O cenário permite então um orquestrador em nuvem, já que o local em que se encontra, também não mais importa, desde que a conexão seja possível.

Esse conceito viabiliza um cenário onde uma máquina, de localização desconhecida, compreenda quais dispositivos existem ao redor de quem irá consumir uma história, e os utilize conforme a necessidade, e se adapte a mudanças bruscas. Isso permite ao consumidor transitar entre ambientes, ou trocar de dispositivos nos quais consome a história naturalmente, e ao autor criar histórias adaptadas ao contexto.

Uma ilustração do ambiente até aqui introduzido pode ser vista na Figura 1. É retratado um cenário onde há inúmeros dispositivos ligados em rede pertencentes a um ambiente doméstico, organizados em sub-áreas. Todos os dispositivos são conectados a uma máquina capaz de os controlar, e consegue compreender onde se encontram. A máquina orquestradora também é capaz de distinguir quem é o usuário, o interator, na figura denominada “Anna”, quais dispositivos possui com ela, e relações com outros interatores. A máquina orquestradora também é capaz de armazenar todos os dados e escolhas de Anna em uma base de dados, na figura denominada “Aura”, que será explicada em tópicos adiante neste trabalho.

Figura 1 - Estrutura de um Orquestrador em Nuvem



Fonte: Elaborado pelo autor (2021)

O criador deve conseguir criar uma história que possa ser contada de formas distintas, em múltiplas mídias, com diversas formas de representar efeitos e detalhes da narrativa, controlando cada pedaço segundo as necessidades mínimas e dispositivos extras, podendo optar ainda por uma linearidade e/ou não de eventos.

Para uma melhor exemplificação do cenário vislumbrado, um fragmento de uma contação de história é apresentado na Figura 2. A interatora Anna está em sua casa (1) e decide iniciar uma contação, utilizando seu celular como o dispositivo para escolher a história desejada (2) a partir de um serviço provedor de histórias. Após a escolha, o orquestrador da contação da história reconhece que Anna dispõe de um *smartphone*, televisão inteligente, caixas de som, e uma assistente doméstica, todos conectados na rede local com acesso à Internet naquele ambiente doméstico. Com os dispositivos disponíveis a contação se adapta, e inicia a apresentação pela televisão (3), utilizando-se das caixas de som na rede para criar uma ambientação (3.1). Após a apresentação do vídeo, a assistente doméstica pede a Anna que dê uma resposta sobre o ocorrido na apresentação (4). Baseado na resposta de Anna (4.2), ações distintas serão tomadas, que impactarão diretamente no resto da história.

Como pode ser observado, a Figura 2 descreve todo um fragmento da contação pensado em ser reproduzido em uma experiência específica para o cenário doméstico, com os dispositivos que ali se encontravam no contexto do interator. A história contada então, pode sofrer variações conforme o ambiente em que se encontra, e dispositivos ali presentes. Caso as caixas de som de Anna não estivessem disponíveis, por exemplo, a contação poderia ocorrer sem problemas, sem

Figura 2 - Contaão de hist3ria simulada em um cen3rio dom3stico - experi3ncia 1



Fonte: Elaborado pelo autor (2021)

gerar varia3es na hist3ria. Contudo, caso Anna se encontrasse em outro lugar, com dispositivos diferentes, uma nova experi3ncia da mesma hist3ria poderia ser proporcionada a ela.

J3 na Figura 3, Anna opta por iniciar a contaao n3o mais dentro de sua casa, mas na rua, em um 3nibus indo ao seu trabalho (1). Ap3s ela selecionar a hist3ria no servio provedor, usando seu celular conectado 3 Internet por uma rede 4G (2), o orquestrador da contaao percebe a localizaao de Anna, e que ali ela n3o possui os mesmos dispositivos do cen3rio descrito na Figura 2. Com isso, o orquestrador opta por uma experi3ncia alternativa para Anna, utilizando apenas o *smartphone* e um fone de ouvido. A contaao que, no cen3rio anterior, fora apresentada em uma TV com caixas de som, 3 apresentada no celular de Anna (3), com sons binaurais para maior imers3o (3.1). Ap3s a apresentaao da cena, na tela do *smartphone*, Anna 3 questionada a escolher uma entre duas op3es (4), e sofrer3 das mesmas consequ3ncias das presentes na figura 2 neste ponto da hist3ria.

3 poss3vel perceber que tanto na Figura 2, quanto na Figura 3, Anna 3 capaz de vivenciar exatamente a mesma hist3ria, mas de formas distintas, vivendo uma experi3ncia pervasiva adaptativa pr3pria ao contexto do qual ela faz parte. Em ambas as experi3ncias de contaao propostas, Anna no final 3 solicitada a interagir. A escolha dessa interaao 3 salva na base de dados de Anna (Aura), que ser3 considerada no momento em que Anna decida continuar a contaao.

A Figura 4 ilustra o caso em que Anna tenha escolhido a op3o "4.2b" em uma das experi3ncias anteriormente descritas, concordando com a personagem

Figura 3 - Contaão de hist3ria simulada em um cen3rio de transporte p3blico - experi4ncia 2



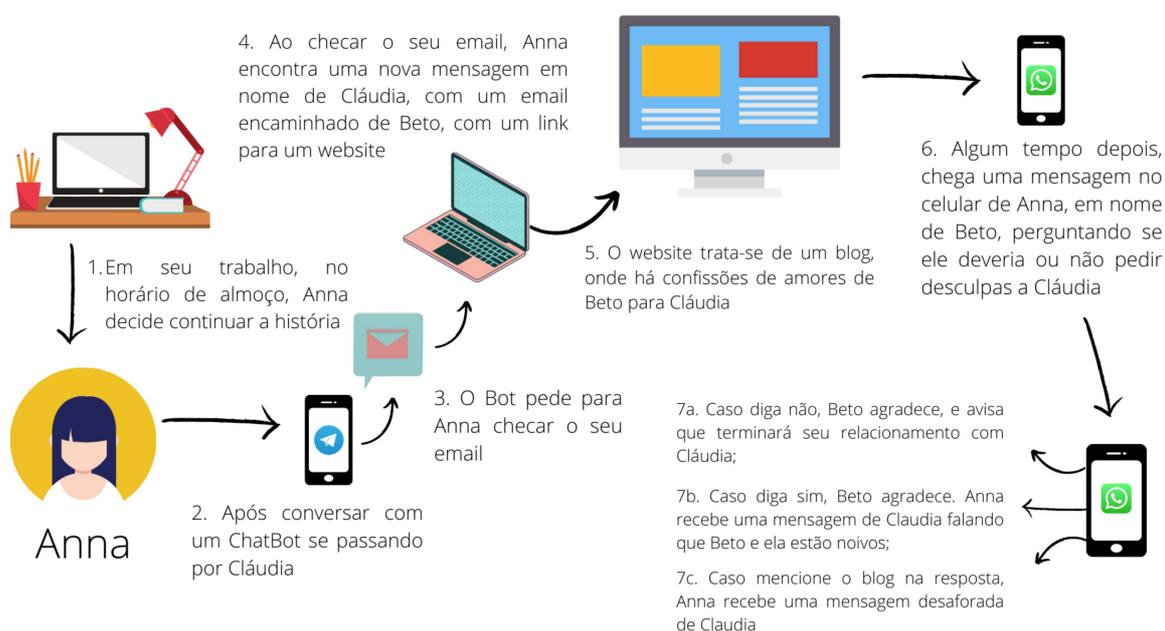
Fonte: Elaborado pelo autor (2021)

“Cl3udia”. Ap3s um per3odo de conversa com o *chatbot* de Cl3udia por um aplicativo de mensagens instant4neas, Anna, j3 em seu trabalho, recebe uma mensagem de Cl3udia, solicitando que a mesma verifique seu correio eletr3nico. H3 ent3o uma mensagem de e-mail de Cl3udia para Anna, com um link para um *website*. Nota-se que informa3es como n3mero de celular ou e-mail de Anna, s3o dados pessoais disponibilizados pela interatora em qualquer momento anterior ao in3cio da contao da hist3ria, como em uma se3o de cadastro no servio do provedor de hist3rias, tornando tais informa3es dispon3veis ao orquestrador.

Ao clicar no *link* fornecido na mensagem de e-mail, Anna consegue acesso a um blog, com mais informa3es pertencentes 3 hist3ria. Posteriormente, Beto, outro personagem da hist3ria, entra em contato por outra aplica3o de mensagens instant4neas de terceiros. A conversa entre Anna e o *chatbot* 4 definitiva para o desenrolar da contao da hist3ria, j3 que a resposta concordando com Beto, discordando, ou citando elementos apresentados anteriormente na contao, determinar3 o desfecho da hist3ria.

Nota-se que em um mundo aberto e pervasivo, sujeito ao livre deslocamento do interator (o pr3prio mundo em que vive), a qualquer momento certas premissas para que a contao possa prosseguir podem ser perdidas. Por exemplo, a conex3o do celular 3 Internet pode se tornar indispon3vel, a visita pr3via a um *website* pode ainda n3o ter acontecido, o interator ainda n3o passou por um local espec3fico no mundo real, entre muitas outras circunst4ncias que poderiam ser pensadas.

Figura 4 - Contaão de hist3ria simulada em um cen3rio de pausa do trabalho



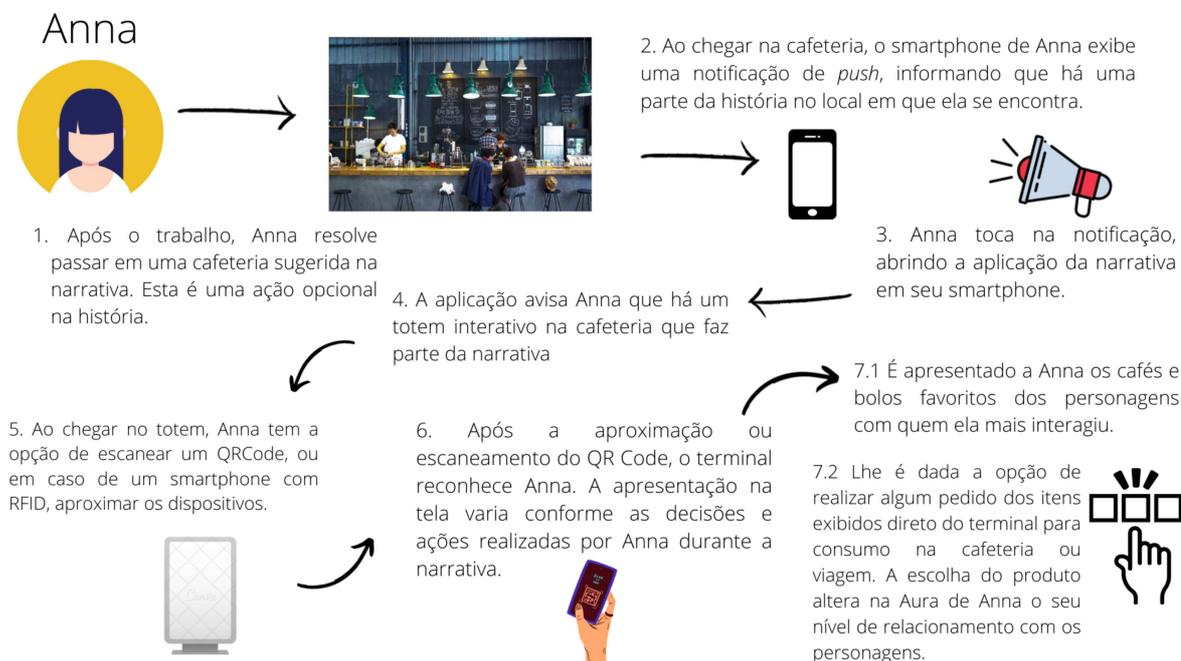
Fonte: Elaborado pelo autor (2021)

De fato, intera3es podem ocorrer diretamente com elementos do mundo real, variando conforme a localiza3o do interator. Para experimentar um ambiente fora de casa, ou locais incomuns a Anna, na Figura 5 Anna opta por ir a uma cafeteria, que o provedor da hist3ria alerta fazer parte da conta3o como um elemento opcional. Contudo, caso Anna opte por concluir um elemento opcional, o mesmo poder3 afetar diretamente no futuro da conta3o personalizada 3 interatora.

Ao chegar na cafeteria, a aplica3o do provedor em seu *smartphone*, por possuir acesso 3 geolocaliza3o (autorizado por Anna), percebe que a interatora se encontra em um local que faz parte da hist3ria. O orquestrador da conta3o ent3o alerta em uma notifica3o por *push* que, naquele local onde Anna se encontra, h3 um totem interativo que faz parte da hist3ria. Ao chegar no totem, Anna pode ser identificada escaneando um c3digo QR, ou mesmo por meio de reconhecimento de sua face. Ap3s reconhecer Anna, o totem exibe informa3es baseadas em todo seu hist3rico na conta3o. No caso, o totem exibe bebidas e comidas favoritas dos personagens com que Anna mais interagiu. O totem permite que a interatora, ent3o, selecione para consumo alguns dos produtos. Os itens que a mesma pedir pelo totem ser3o armazenados como informa3es de decis3o de Anna, e ser3o utilizados futuramente para desbloquear ou n3o novos fragmentos da conta3o que ela experimentar3.

Fragmentos da Hist3ria de Anna constru3dos segundo o modelo proposto neste trabalho podem ser encontrados no Ap3ndice D.

Figura 5 - Contaço de história simulada em um cenário de ambiente externo



Fonte: Elaborado pelo autor (2021)

2.3.1 Requisitos para um Modelo de Representação da Contaço Pervasiva de Histórias

Tomando como base as situações e possibilidades descritas nos cenários de uso apresentados na seção anterior, foi possível levantar os seguintes requisitos para a concepção de um novo modelo de representação da contaço pervasiva de histórias e máquina capaz de representar o modelo:

R1: [Modelo] Possuir alto nível de abstração, de forma a permitir a prototipação de ferramentas de autoria visuais com mapeamento próximo ao modelo, que seja facilmente entendido por profissionais do domínio de criação de conteúdos hipermídia;

R2: [Modelo] Tornar uniforme a interface de E/S com dispositivos multimídia, sensores e atuadores IoT. Da mesma forma, a interface deve eximir o autor de especificar a descoberta, comunicação e comportamento dos dispositivos;

R3: [Máquina] Reunir dispositivos multimídia e IoT tanto do interator quanto aqueles cuidadosamente instalados pelo autor ou pelo provedor da história para possível participação na contaço;

R4: [Modelo e Máquina] Suportar a adaptabilidade da contaço de história, sua personalização, segundo o perfil do interator e de todo seu contexto atual;

R5: [Modelo e Máquina] Suportar as variabilidades de ambientes pervasivos (desconexões, mobilidade, mudanças de foco do interator, desistências);

R6: [Modelo] Guiar a contaço de uma história pelas circunstâncias de

contexto do interator e não somente pelo sincronismo de mídias e interações diretas;

R7:[Modelo] As circunstâncias incluem não só o contexto do interator, mas também o contexto da própria história, que não necessariamente é baseada em eventos ordenados apenas;

R8: [Modelo] Rastreamento do progresso do interator na história, criando um histórico que marque as experiências concluídas, as decisões tomadas, dados relevantes recolhidos.

R9: [Modelo] Permitir o acesso ao histórico do interator tanto na história corrente como e em outras histórias;

R10: [Modelo] Permitir a segmentação das histórias em fragmentos que demarcam semanticamente as experiências disponíveis ao interator. Segmentação também em termos de episódios pode ser interessante, dada a possibilidade de longos enredos em um mundo aberto e pervasivo;

R11: [Modelo e Máquina] Suportar experiências com a participação coordenada de múltiplos interatores;

R12: [Máquina] Auxiliar na separação de interesses entre orquestração e dispositivos de E/S, vislumbrando um orquestrador onipresente e que por si não está necessariamente incorporado a um dispositivo principal do usuário.

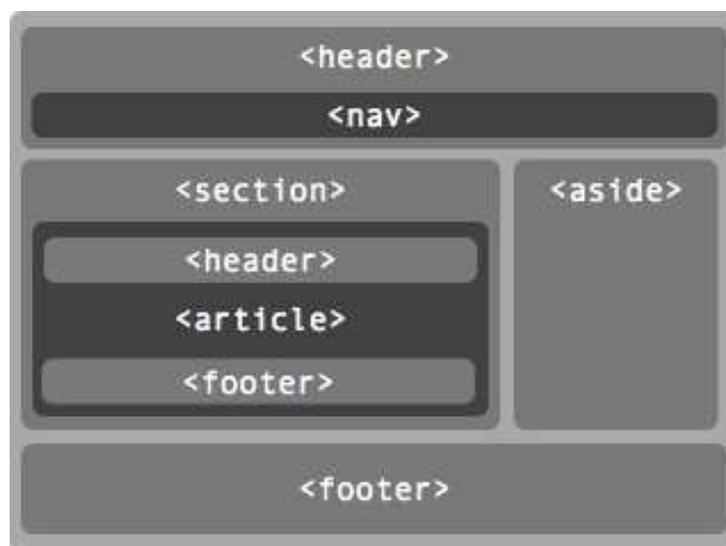
3 TRABALHOS RELACIONADOS

Este capítulo aborda alguns trabalhos relacionados que serviram de base para o presente estudo. O levantamento foi conduzido com o intuito de compreender e possuir um panorama das linguagens e modelos, da área de hipermídia, multimídia e mulsemídia no âmbito de representação de informação para execução de apresentações, e vislumbrá-las no cenário de construção de narrativas pervasivas.

3.1 HTML5

O HTML (***HyperText Markup Language***) surgiu inicialmente na década de 1990, como uma linguagem de marcação para construção de páginas de texto na *web*, com hiperlinks entre páginas para navegação. Sua evolução vem acompanhando novas necessidades de consumo, entre elas a de conteúdo multimídia transmitido pela rede, tal qual o fenômeno de serviços de *streaming*, e aplicações cliente-servidor, com clientes executando no contexto de navegadores. Atualmente, a quinta versão do HTML, o HTML5 surgiu com novas tecnologias para modernizar o padrão da web (62).

Figura 6 - Estrutura do corpo (*body*) de um documento HTML5



Fonte: Rodrigo Aramburu (5)

O HTML5 conta nativamente com novos elementos para conteúdo multimídia, além de elementos de primeira classe para áudio e vídeo (62), sem mais a necessidade de contar com *plugins* externos, comuns na década de 2000 como “Adobe Flash Player”. Também foi criado um elemento para desenvolvimento gráfico, o “Canvas”, sendo este um “contêiner”, para o uso de JavaScript(EcmaScript) (36) para a renderização final de elementos 2D e 3D (64). Para estilização, responsividade e personalização, conta com suporte a estilos em cascata pelo CSS3 (63).

Com a manipulação de sua estrutura através da DOM (*Document Object Model*), uso de *scripts*, tal qual JavaScript(EcmaScript) com constante acesso a servidores, aplicações robustas e adaptativas, executando em grande parte no cliente, fazem parte de milhares de serviços *web* contemporâneos.

A quinta versão segue a mesma estrutura básica de sua versão anterior. A primeira linha define a categoria do documento que o navegador deverá interpretar, sinalizada pela *tag* `<!DOCTYPE html>`. Posteriormente, dois elementos são obrigatórios em todo documento HTML, `<head>` e `<body>`. No cabeçalho (`<head>`) são inseridos metadados da página, sendo esses muito utilizados por buscadores e sistemas automáticos com uso de funcionalidades como *web scraping* (35). No mesmo elemento é possível definir e/ou referenciar estilos através do *Cascading Style Sheets* (CSS3), e *scripts* como ECMAScript.

No corpo (`<body>`) define-se a semântica do documento. Diferente de sua versão anterior, o HTML5 possui diversos elementos de primeira classe para semânticas distintas como `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`, `<p>` (*paragraph*), conforme ilustrado na Figura 6. Em suma, cada um deles define um tipo diferente de bloco de conteúdo e são organizados em uma hierarquia capaz de estruturar melhor o documento, facilitar a criação de leiautes e ajudar na extração de semântica.

Apesar das novas *tags* para vídeo e áudio possibilitarem a configuração de controles básicos de reprodução de mídias contínuas, ainda não é possível definir as diversas formas de relações entre mídias e sincronismo, conforme as relações de Allen (2). Para tais fins, e para propiciar as atuais reproduções adaptativas, faz-se necessário uso de *scripts* que, segundo Fressz (17), se configura, de fato, uma abordagem imperativa.

Em contrapartida, o uso intensivo de *scripts* para possibilitar maior funcionalidade dificulta a extração da semântica e relacionamento entre elementos, além de exigir cada vez mais a necessidade de profissionais especializados com conhecimentos obrigatórios em múltiplas tecnologias para produção de documentos, incluindo *frameworks* avançados como React e VUEjs para produções de aplicações em larga escala.

Sendo assim, apesar da versatilidade do HTML para construção de aplicações e ferramentas web, o mesmo torna-se extremamente dependente de *scripts* para tais produções. Para a construção de narrativas pervasivas, onde há uma extrapolação das mídias convencionais, devido a falta de suporte nativo a dispositivos, múltiplos usuários, armazenamento de dados do usuário e compartilhamento, e a necessidade de execução em um navegador, a linguagem torna-se limitada para uso em narrativas pervasivas, uma vez que necessitaria de *scripts* para possibilitar a orquestração e execução.

3.2 NCM - NESTED CONTEXT MODEL (3.0)

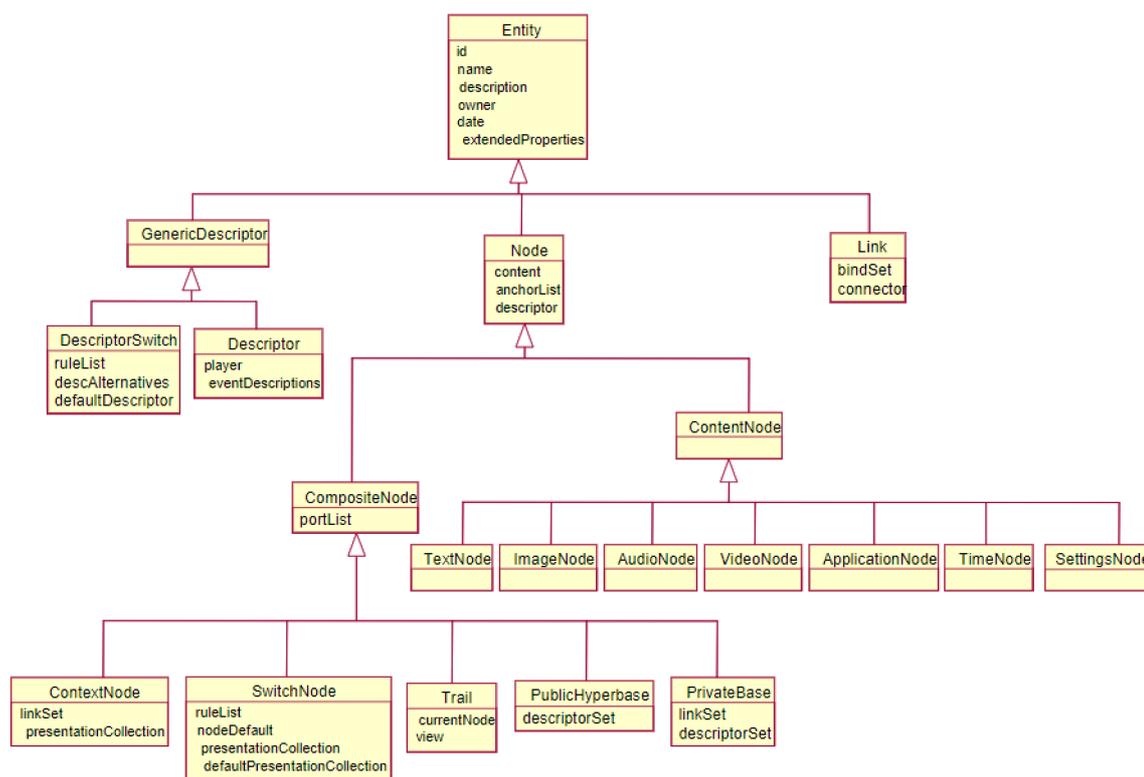
O NCM, atualmente na versão 3.0, é um modelo conceitual voltado a hipermídia (55). O modelo visa representar conceitos estruturais, eventos e relacionamento entre mídias, definindo também regras e operações para manipulação, sincronismo, reprodução e atualização das estruturas e documentos.

NCM é baseado no conceito de nós (*nodes*) e elos (*links*). Um Nó representa fragmentos de informação, e Elos a relação entre os Nós. Nós podem representar conteúdo em si ou serem uma composição de outros Nós. Elos estabelecem relacionamentos de causalidade ou restrição entre os Nós. Eles se apoiam em Âncoras para estabelecer os relacionamentos, já que elas delimitam um subconjunto das informações de um Nó, no tempo ou no espaço (54).

Nós de conteúdo podem ter atreladas propriedades (*Property*) da mídia, como, por exemplo, posição espacial, proporção e cor de fundo. Através da interface (*Interface*), é possível também definir relacionamentos que envolvam propriedades e âncoras (que também são um tipo de interface).

A forma de exibição, ou seja, o controle da mídia com o tempo e espaço, além de onde será exibida, pode ser definida através de descritores (*Descriptor*). Dessa forma, o modelo permite a reprodução distribuída, explorando múltiplos dispositivos. Cada nó de conteúdo é associado a um descritor, e, assim, ele fica atrelado a um grupo específico de dispositivos (classe de dispositivo) pré-definido (56).

Figura 7 -Visão geral da hierarquia de classes do NCM



Fonte: Soares (55).

O NCM permite a adaptabilidade da apresentação e do conteúdo. Através das regras (*rules*), variações do conteúdo já pré-definido (*DescriptorSwitch*) podem ser chamados conforme a lógica/regra estipulada, como, por exemplo, um áudio em uma segunda língua.

A sincronia de reprodução de múltiplas mídias é estipulada pelo uso de conectores. Uma vez definido pelo conector, é possível que uma ação “x” seja executada no mesmo momento em que “y” é iniciada, criando o evento de casualidade em uma apresentação. Para realizar a ligação entre conectores e interfaces, cria-se um elo (*link*), permitindo assim relacionamentos de sincronismo tanto temporal, quanto espacial (54).

Por fim, uma grande contribuição do NCM advém da possibilidade de reuso e agrupamento. O modelo permite a criação de conjuntos de apresentações, com mídias, relações, links e afins, como um contêiner. Tais nós de composição podem ter relacionamentos com outros nós, de outros conjuntos, permitindo ainda um aninhamento de composições.

O NCM possui diversos elementos capazes de propiciar a construção de uma narrativa pervasiva, provendo execuções baseadas em eventos, nós genéricos, variações de conteúdo, armazenamento de dados e afins. Contudo, algumas modificações se fazem necessárias para melhor se adaptar ao ambiente pervasivo, como efeitos sensoriais, múltiplos usuários, armazenamento e acesso a histórico de narrativas distintas, orquestração em nuvem, entre outras. Tais inúmeras mudanças poderiam aumentar drasticamente a complexidade do modelo, uma vez que o mesmo fora projetado para outro tipo de cenário.

3.3 NCL - NESTED CONTEXT LANGUAGE

NCL é uma linguagem declarativa XML, baseada no modelo conceitual NCM, para fins de desenvolvimento de aplicações multimídia interativas, com forte distinção entre conteúdos de mídia e estrutura da aplicação, sem restrição a categorias de conteúdo dos objetos de mídia de uma aplicação (54).

A princípio, a linguagem foi desenvolvida com a premissa de ser utilizada na Web, assim como o HTML. Contudo, no Brasil, a linguagem ganhou grande representatividade junto ao desenvolvimento do *middleware* Ginga como padrão do Sistema Brasileiro de TV Digital Terrestre (1), devido à necessidade de uma linguagem declarativa padronizada para o desenvolvimento de aplicações interativas para TVs digitais no país.

A estrutura básica de um documento NCL, versão atual 3.0, consiste em um elemento raiz primário, `<ncl>`, com dois filhos `<head>` e `<body>`. O cabeçalho `<head>` é composto por descritores, que definem como as mídias serão apresentadas, regiões, que indicam posicionamento da mídia durante a apresentação, conectores causais, responsáveis por definir semântica de relações a serem usadas em elos (links), além de metadados, e regras para adaptação de conteúdo. No corpo `<body>`, são definidos os objetos de mídia `<media>` (nós de conteúdo), elos `<links>`, portas `<port>` e nós de composição dos tipos contexto `<context>` e alternativas `<switch>`. Os nós de conteúdo utilizados na aplicação possuem um conjunto de propriedades `<property>`, que facilitam a customização de aspectos relacionados à apresentação dos objetos de mídia que representam, permitindo também lidar com *scripts*. Os elementos da linguagem que definem a estrutura do documento NCL podem ser observados na Figura 8. Comentários no código indicam em que partes da estrutura

devem figurar os demais elementos.

Figura 8 - Estrutura básica de um documento NCL

```
<ncl id="exemplo" xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile">
  <head>
    <ruleBase>
      <!-- regras para seleção de objetos e descritores -->
    </ruleBase>
    <regionBase>
      <!-- regiões onde as mídias serão apresentadas -->
    </regionBase>
    <descriptorBase>
      <!-- definição de como as mídias serão apresentadas -->
    </descriptorBase>
    <connectorBase>
      <!-- definição de como as mídias serão apresentadas -->
    </connectorBase>
  </head>
  <body>
    <port id="nomeDaPorta" component="umNoDesteContexto" />
    <!-- contextos, nós de mídia e suas âncoras, elos e outros elementos -->
  </body>
</ncl>
```

Fonte: Freesz Júnior 2017 (18)

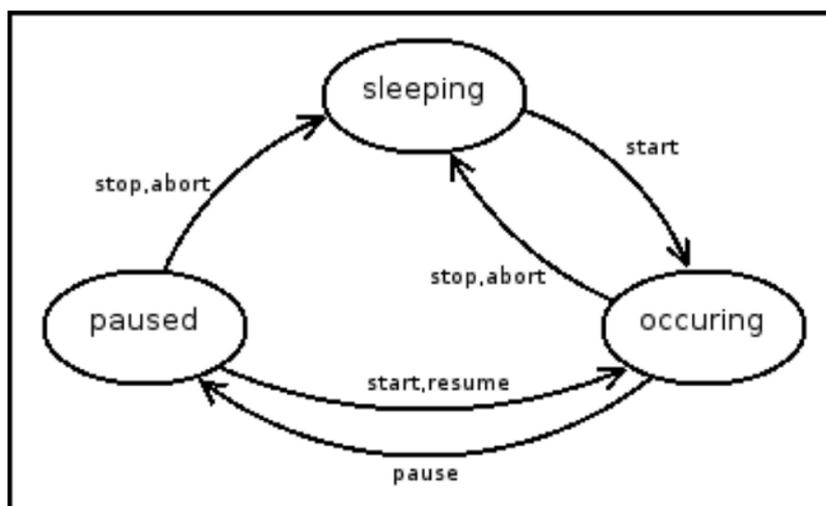
A linguagem permite a manipulação de quatro tipos de eventos atrelados a nós de conteúdo e de composição, envolvendo suas interfaces: apresentação, seleção, atribuição e preparação. Cada nó possui, para cada tipo de evento, uma máquina de estados associada. Elas possuem em comum três estados e cinco transições, conforme ilustrado na Figura 9. Tomando o evento de apresentação como exemplo, em um primeiro momento, a máquina de estados fica em espera (estado *sleeping*), aguardando que a apresentação do nó seja iniciada (transição *start*), a qual leva a máquina ao estado ocorrendo (*occurring*). Uma vez ocorrendo, a apresentação poderá voltar para seu estado inicial de espera se for abortada ou encerrada (transições *abort* e *stop*, respectivamente), ou poderá ser pausada (transição *pause*). Uma vez pausada (estado *paused*), a apresentação poderá ser retomada (transições *start* ou *resume*), ou ser finalizada (com transição *stop* ou *abort*), voltando ao estado de espera (*sleeping*).

NCL permite a especificação de módulos, capazes de estender a linguagem para necessidades mais específicas de aplicações multimídia. Algumas extensões serão observadas nos tópicos a seguir, como suporte a Múltiplos Usuários, Interações Multimodais e Efeitos Sensoriais.

3.3.1 NCL Estendida para Suporte a Multiusuários e Interações multimodais

Uma interface comum consegue obter uma única modalidade de entrada de informações do usuário por vez, sendo ela obtida por sensores que captam e processam ações humanas, como voz, som, toque, etc. Conforme explicado por Guedes et al. (21), uma interface de usuário multimodal (*Multimodal User Interface*) consiste na capacidade de se obter múltiplos *inputs* de várias modalidades (como

Figura 9 - Máquina de estados de eventos NCL.



Fonte: PUC Rio, TeleMidia Lab (48)

som, imagens, toque, e afins). Tais informações, por sua vez, podem ser geradas por mais de um usuário, e/ou grupo de usuários. Aplicações capazes de obter dados desses múltiplos usuários possuem uma interface denominada interface de múltiplos usuários (Multiuser Interfaces). Tais interfaces devem distinguir indivíduos e suas respectivas ações.

Interfaces multimodais-multiusuários, apesar de ainda rudimentares, constituem parte da realidade plausível em dispositivos e sistemas contemporâneos. No nível atual das tecnologias multissensoriais, junto à facilidade de obtenção desses novos sensores por usuários finais, os trabalhos de Guedes et. al. (19), (20) e (21) abordam a necessidade de linguagens multimídia contemporâneas (como NCL e HTML), de evoluírem e criarem abstrações para conseguirem capturar e lidar com várias informações multimodais e distinguir de quais usuários e/ou grupos que elas advêm.

Em seu trabalho, Guedes et al. (21) apontam o fato de, por serem possíveis inúmeros cenários de ambientes, com múltiplas interações, possuindo equipamentos e interatores distintos, um modelo conceitual multimodal-multiusuário deve possuir um acoplamento fraco entre estrutura e usuários. De tal modo, o modelo torna-se capaz de permitir uma flexibilidade de especificação, mantendo uma mesma estrutura para definir entrada e saída, apenas se adequando ao efeito de interação pretendido a cada usuário.

Linguagens multimídia mais populares como HTML, SMIL e NCL, são linguagens declarativas, com foco em interação de um único usuário, com acoplamento forte a estruturas, restringindo-se a entradas convencionais, como teclado, mouse, ou controle da TV. Apesar de NCL especificar apresentações multimídia interativas, a versão atual (3.0) não permite que autores criem aplicações para múltiplos usuários adequadamente. Além disso, NCL trabalha de forma a suportar múltiplos

dispositivos apenas como uma extensão do dispositivo primário em que reproduz a apresentação.

De modo a melhorar a categorização de modalidades de interação, Guedes et al. (19) definem três cenários de interação para Modelos de Autoria Multimodal e Multiusuário, denominados “Put-That-There”, “I-Get-That-You-Put-It-There” e “Anyone-Get-That-Someone-Else-Put-It-There”.

“Put-That-There” é uma extensão do trabalho de Bolt (11), onde a interface deve prover, para um único usuário interagindo com o sistema, diferentes modalidades de entrada capazes de manipulação de um objeto virtual simultaneamente, e ter sua saída representada por um ou mais dispositivos de saída distintos.

Em “I-Get-That-You-Put-It-There”, a tarefa deve ser realizada por dois usuários distintos, sendo o sistema capaz de distinguir os usuários e de permitir a múltipla interação, tal qual ocorre em jogos eletrônicos, onde há o jogador 1 e o jogador 2.

“Anyone-Get-That-Someone-Else-Put-It-There” é similar ao cenário anterior, com a diferença de que uma primeira interação é feita por um usuário e a posterior pode ser definida pelo sistema para que um segundo usuário a dê continuidade. Este evento ocorre similarmente a jogos que alternam os momentos em que cada jogador deve realizar as ações.

Para permitir que o NCM (e por consequência a NCL) possuam os recursos de multiusuários e multimodais citados a cima, Guedes et al. (19) propõem as seguintes extensões:

- Duas novas subclasses de *contentNode*. *MediaNode*, usada para representar modalidades de saída tanto de mídias tradicionais, quanto não convencionais, e *RecognitionNode*, para representar somente a modalidade de entrada, estendendo a múltiplas categorias de *inputs*. Assim, os eventos de interação não se limitam às entradas convencionais (teclas e mouse) e são definidos por categoria de entrada, não de dispositivo (e.g. *SpeechRecognitionNode*);
- Uma variação do elemento âncora (*anchor*), capaz de lidar com eventos de “reconhecimento”. Caso eventos definidos no nó de reconhecimento (*recognitionNode*) ocorram, as *RecognitionAnchor* são acionadas;
- Uma nova classe responsável por representar grupos de usuários, permitindo um controle pelo autor da representação e interação entre usuários distintos, *UserClass*;
- E por fim, um novo atributo capaz de relacionar diretamente um usuário específico ao papel de eventos em *Links*.

Para Nigay e Coutaz (42), as interfaces de múltiplos usuários devem possuir três combinações de interação; redundante, quando apenas uma entre as interações definidas devem ocorrer; complementar, quando todas as interações definidas devem ocorrer; e sequencialmente complementar, quando além de todas terem de ocorrer, precisa-se que ocorram em ordem. Para que o NCM contemplasse as três modalidades citadas, Guedes (19) propõe um novo elemento denominado

RecognitionNodes, adjunto ao elemento *CompoundCondition*, usando operadores já definidos no NCM, como: “or”, satisfazendo o arquétipo de Redundante; operador “and” para Complementar, e um novo operador chamado “seq”, para definir que os itens deverão ocorrer obrigatoriamente em ordem.

NCL passa, então, a refletir as modificações feitas no NCM, criando elementos, <media> como *MediaNode*, <input> como *RecognitionNode*, <userClass> como *userClass*. No que lhes concerne, os elementos já existentes em NCL <bindParam> e <linkParam> são estendidos com os atributos de componente e interface, e o <connectorParam> é estendido com os atributos *eventType* e *attributeType*, conforme pode ser observado na Figura 8.

Figura 10 - Elementos NCL relacionados às extensões NCM propostas por Guedes (19).

NCL element	Main attributes	Content
media	id, src, type, descriptor	(area property)*
input	id, src, type	(area property)*
connectorParam	name, eventType, attributeType	empty
bindParam	name, value, component, interface	empty
linkParam	name, value, component, interface	empty
userClass	id, min, max, userClassDescription	empty

Fonte: Guedes et al (2016) (19).

Trabalhos mais recentes como de Barreto (7), propõem interações multimodais focadas em TV Digital Interativa para a nova versão proposta de NCL, denominada NCL 4.0 (7). Esta versão introduz suporte a eventos de interação multimodais e de múltiplos usuários, com a capacidade de identificar o usuário que interagiu com o aplicativo de TV Digital Interativa.

3.3.2 NCL Estendido para Suporte a Efeitos Sensoriais

Conforme descrito no início da Seção 3.3, para a representação de um item midiático em NCL, utiliza-se do elemento <media>. Esse elemento não define explicitamente o tipo de mídia a ser executada, sendo de possível definição através do atributo *type*, seguindo o formato MIME Media Types. Soares (54), no que lhe concerne, não apresenta representações para efeitos sensoriais, retendo aos tipos de mídia convencionais áudio, vídeo, texto, imagem e aplicação, elementos comuns na época em que seu estudo fora desenvolvido.

De modo a permitir o uso de efeitos sensoriais e facilitar a autoria de aplicações *multemídia* com NCL, Rodrigues (15) propõe estender a linguagem, baseando-

se na arquitetura de representação de informações para comunicação entre aplicações e equipamentos de reprodução descritos pelo padrão MPEG-V (24).

Para evitar novas *tags* XML em NCL apenas para representar os efeitos sensoriais, são propostos novos tipos para o atributo *type* do elemento `<media>`. Na representação criada, a palavra referente ao efeito sensorial especificado pelo MPEG-V deverá ser sempre precedida por “application/x-sensory-effect-”. Como exemplo, um efeito sensorial do tipo vento, *WindType*, deverá ser representado como “application/x-sensory-effect-WindType”.

Para permitir as configurações de propriedades de mídias de efeitos sensoriais a serem reproduzidos, o elemento `<property>`, filho de `<media>`, passa a suportar atributos análogos às propriedades de efeitos sensoriais especificadas pela ISO/IEC 23005-3:2019 (24). Além de `<property>`, também é possível utilizar os descritores para a definição de atributos de objetos de efeitos sensoriais, com a vantagem do mesmo poder ser reaproveitado em diferentes objetos de mídia, desde que compartilhem de informações em comum.

Listing 3.1 – Exemplo da tag *media* para efeitos sensoriais com propriedades

```
<media id="media\_3" type="application/x-sensory-effect-WindType"/>
  <property name="location" value="left:top:*/>
  <property name="intensity" value="0.3"/>
</media>
```

A Listagem 3.1, retirada de Rodrigues (15), mostra um exemplo de objeto de mídia de efeito sensorial do tipo vento, com as propriedades de posição do efeito sensorial, *location*, a intensidade, *intensity*, ambas no formato definido pelo padrão MPEG-V.

O elemento `<region>` utilizado para atributos de posicionamento, inicialmente, se adequa somente a elementos bidimensionais, visto que fora pensado para mídias representadas em expositores bidimensionais (2D). Como efeitos sensoriais podem possuir ação em três dimensões, realiza-se uma extensão dos atributos de `<region>`, para contemplar representações tridimensionais (3D), sem necessitar a criação de um novo elemento. Com base no MPEG-V, foram incluídos os atributos *x-axis*, *y-axis* e *z-axis*, para definir a propriedade *location* de objetos de efeitos sensoriais. É possível também definir por coordenadas polares, acrescentando o atributo *polar* para ângulo polar, e *azimuthal*, para ângulo azimutal, necessitando o uso adjunto dos atributos *width* e *height* (largura e altura), mas tendo seus valores definidos com o sufixo *deg* (também aplicado ao ângulo azimutal e polar), para indicar que os valores são definidos em graus.

Em um trabalho mais recente, Josué (28) visa evitar problemas relacionados a sincronização durante a apresentação de aplicações de cunho mulsemídia, propondo uma operação para preparação de objetos de mídia e efeitos sensoriais. Assim, a máquina de apresentação pode realizar uma preparação automática de efeitos sensoriais conforme o tipo do efeito, condições de rede e comportamento temporal da aplicação. O trabalho de Josué (28) estende NCL e contribui para a versão 4.0 da linguagem (7), sendo inclusive capaz de utilizar componentes padronizados pelo MPEG-V.


```

<par endsync="vid1">
  <video xml:id="vid1"/>
  <excl dur="indefinite">
    <par begin="englishBtn.activateEvent">
      <audio begin="vid1.begin" src="english.au"/>
    </par>
    <par begin="frenchBtn.activateEvent">
      <audio begin="vid1.begin" src="french.au"/>
    </par>
  </excl>
</par>
<seq>
  
  
  
  <!-- etc., etc. -->
</seq>
...
</smil>

```

O modelo também permite adaptação de conteúdo para caso alguma condição predefinida seja verdadeira, permitindo assim apresentações adaptativas e personalizadas através do elemento <switch>. A Listagem 3.4 exemplifica o elemento.

Listing 3.4 – Exemplo de uso do elemento *switch* em SMIL

```

...
<par>
  <video src="anchor.mpg" .../>
  <switch>
    <audio src="dutchHQ.aiff" systemBitrate="56000"... />
    <audio src="dutchMQ.aiff" systemBitrate="28800"... />
    <audio src="dutchLQ.aiff".../>
  </switch>
</par>
...

```

3.5 MODELO STorM

Conforme definido por Freesz (18), STorM (Scene- and Track-oriented hypermedia Model) é um modelo conceitual hipermídia, com entidades de alto nível de abstração, focado em usuários de domínio específico da área de comunicação, para criação de apresentações para mídias do tipo “Digital Out-of-Home” (DOOH), também conhecidas por mídias de Sinalização Digital (Digital Signage). Mídias DOOH são comumente utilizadas para exibição pública, como propagandas em locais diversos como transportes públicos, centros comerciais, ruas e ambientes corporativos.

Figura 11 - Exemplo de aplicação de mídias em DOOH em ambiente urbano.



Fonte: Lukas Hartmann.

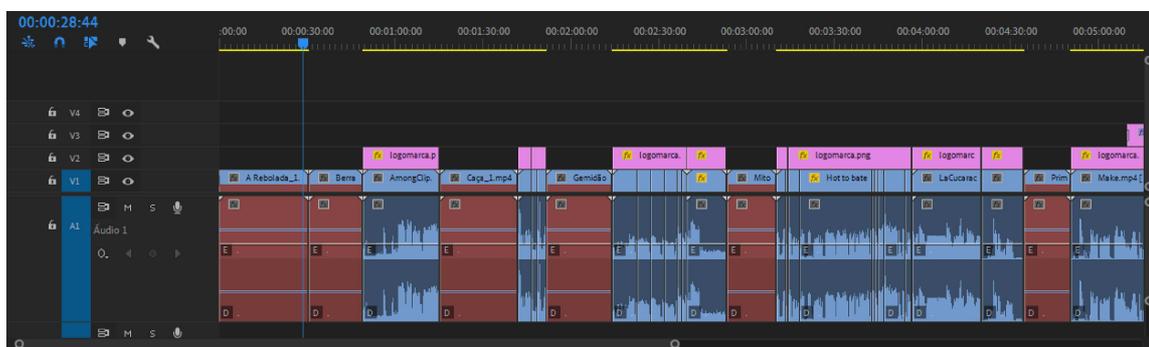
STorM define, como pilares de sua organização do documento multimídia, as abstrações de “cenas” e “trilhas”, pensadas para se assemelharem a conceitos próximos a existentes na indústria audiovisual. Na Figura 12, capturada de um programa popular de edição de vídeo, é possível constatar o uso do conceito de trilhas de vídeo e áudio, compostas por fragmentos de mídias editadas, todas dentro de uma Cena em edição.

Cenas são um conjunto de objetos definidos em paralelo, sendo as Cenas similares ao conceito de “contextos” de NCL, em que se define o escopo da apresentação, composto por mídias e contextos aninhados. Já as trilhas são similares ao conceito da composição “seq” do SMIL. As trilhas executam em ordem (sequencialmente) os objetos das cenas que ali foram definidos, com a duração da trilha podendo ser baseada na duração total cumulativa dos objetos internos, ou determinada nas configurações.

As trilhas e seus objetos também possuem propriedades, capazes de definir relacionamentos durante a exibição da cena, como região de exibição, delay, ou volume da trilha, conferindo maior personalização.

O STorM possui uma variação de trilhas que quebram a sequência temporal convencional, as denominadas trilhas interativas. As trilhas interativas permitem, através da entidade elo (links), que um objeto durante a execução aponte para um objeto externo da trilha atual, gerando a execução da mesma. A trilha interativa

Figura 12 - Fragmento do Premiere Pro de uma cena composta por trilhas com seus respectivos objetos de mídias.

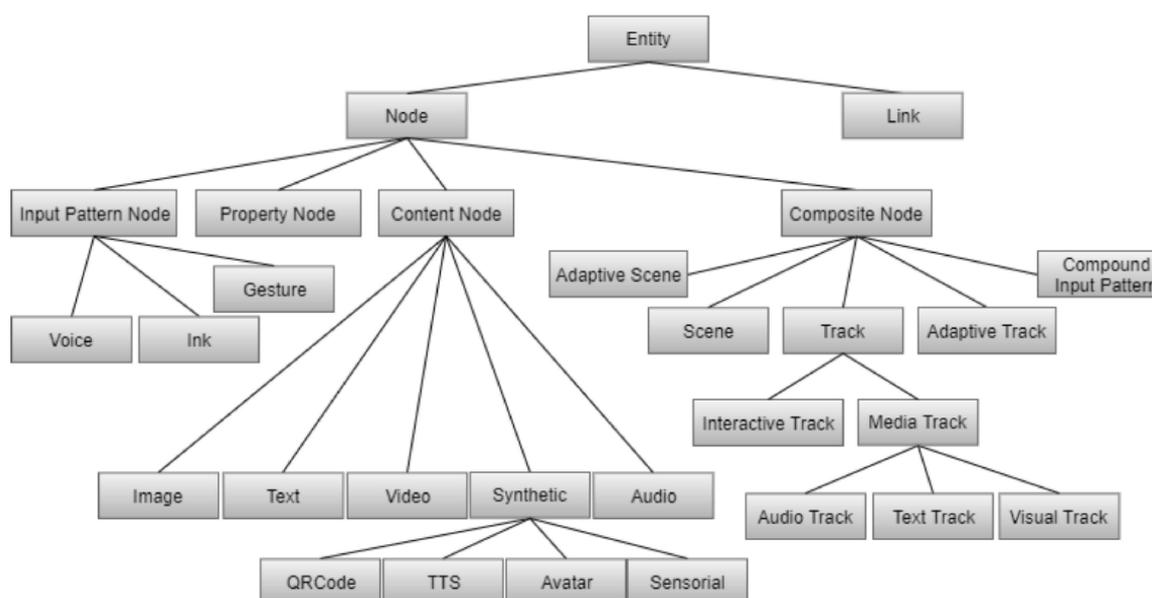


Fonte: Elaborado pelo autor (2021).

permite apontar também para aplicações auto contidas chamadas *widjets*.

Tanto as Cenas quanto as Trilhas podem ser adaptativas. O modelo permite que o autor declare caminhos alternativos para caso alguma trilha ou cena seja impedida de tocar, ou alguma preferência do usuário, permita que outra seja tocada. Permite-se assim que somente uma parte do todo sofra variações, não necessitando de um novo documento para cada variação desejada pelo autor.

Figura 13 - Entidades do Modelo STorM.



Fonte: Freesz Junior (2017)(18).

STorM pode nativamente obter informações de dispositivos de entrada, por meio de *input nodes*, posicionados em trilhas ou cenas, como um nó de mídia

qualquer. Em trilhas, o padrão de entrada é aguardado antes de prosseguir a reprodução. Sendo assim, um objeto de entrada é representado tal qual uma mídia com um link associado, que será interpretado quando o objeto receber a entrada.

Conforme pode ser observado na Figura 13, contendo as principais entidades do modelo, o modelo STorM possui bastante influência do NCM, com entidades em comum como nós e nós de conteúdo (*content node*) e suas respectivas mídias, links, e afins. Contudo, o modelo STorM garante nativamente entidades de interação e adaptação, que foram de grande influência no modelo PST.

3.6 LINGUAGEM STorML

STorML é uma meta-linguagem XML do modelo STorM. O elemento raiz chama-se `<storml>`, que possui um identificador único, tendo como filhos `<head>` e `<body>`, que tem princípios similares a de outras *tags* com a mesma nomenclatura em linguagens declarativas como HTML. O elemento `<head>` contém metadados do documento, leiautes e padrões de entrada usados em relacionamentos e interações multimodais existentes no elemento `<body>`. O corpo (`<body>`) contém os elementos que definem a apresentação. O corpo possui o campo *presentation-Mode* que permite determinar se todas as apresentações internas serão sequenciais ou aleatórias, permitindo ou não o uso de links para permitir o uso de objetos interativos.

Todos os módulos do STorML apresentam o elemento `<meta>`, sendo esse responsável pela descrição do documento, para facilitar buscas, recuperação e reuso de elementos em uma possível ferramenta gráfica de autoria.

A Figura 14, proposta por Freesz (18), apresenta o diagrama de elementos presentes na linguagem STorML, fornecendo uma visão geral da organização da linguagem e seus principais elementos.

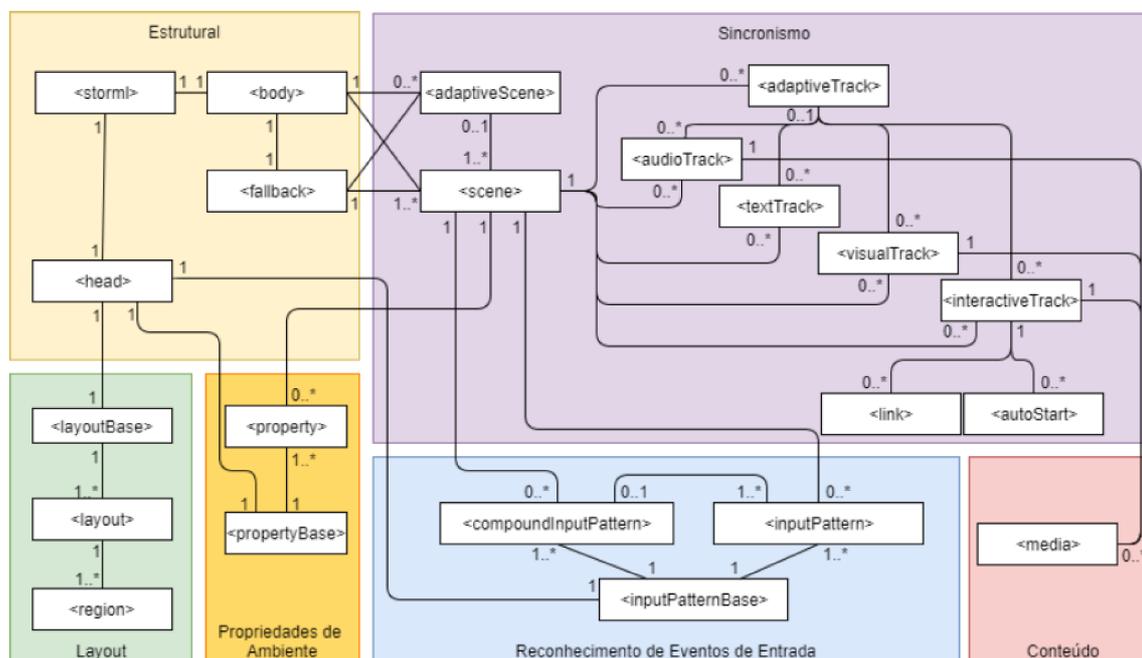
No corpo ficam as cenas (`<scene>`) citadas na Subseção 3.5. Além das cenas, há também as cenas adaptativas (`<adaptiveScene>`) para declaração de cenas substitutas caso alguma propriedade válida permita a execução de uma variação de cena. Também no corpo, encontra-se o elemento `<fallback>`, que pode ser utilizado para cenas de *backup*, em caso de indisponibilidade de rede.

No cabeçalho (`<head>`), o elemento `<layoutBase>` contém os leiautes da apresentação, sendo o filho `<layout>` responsável por especificar formas de exibição, podendo definir formas distintas de representar as mídias, tanto no dispositivo principal quanto em dispositivos secundários.

Para realizar a sincronia e a orquestração da apresentação, existem duas *tags* principais, cena (`<scene>`) e trilha (`<track>`), sendo esta última um filho da cena. As cenas são executadas de forma sequencial dentro do `<body>`, já as trilhas são executadas paralelamente entre si. Os elementos internos das trilhas são executados sequencialmente, um após o outro.

As trilhas em uma cena são, então, executadas simultaneamente conforme a programação, permitindo que trilhas como de vídeo ou imagem (`<visualTrack>`) executem ao mesmo tempo que uma trilha de áudio (`<audioTrack>`), e uma trilha

Figura 14 - Diagrama de elementos da linguagem STorML.



Fonte: Freesz Junior (2017) (18).

de texto (`<textTrack>`), como legenda, por exemplo. As trilhas possuem semântica de sequência, ou seja, todas as mídias declaradas internamente a uma trilha, são executadas uma após o término da outra, seguindo a relação “A meets B” de Allen (2).

Outras variações de trilha são as trilhas interativas (`<interactiveTrack>`) que permitem qualquer tipo de mídia ou elo seja executada, como aplicações ou links. Também há cenas e trilhas adaptativas (`<adaptiveScene>` e `<adaptiveTrack>`), onde é possível definir propriedades para execução das mesmas, para que o *player* avalie se a mesma pode ou não ser executada quando chamada. Em caso de múltiplas cenas ou trilhas alternativas, a primeira que cumprir os requisitos é acionada e as demais ignoradas.

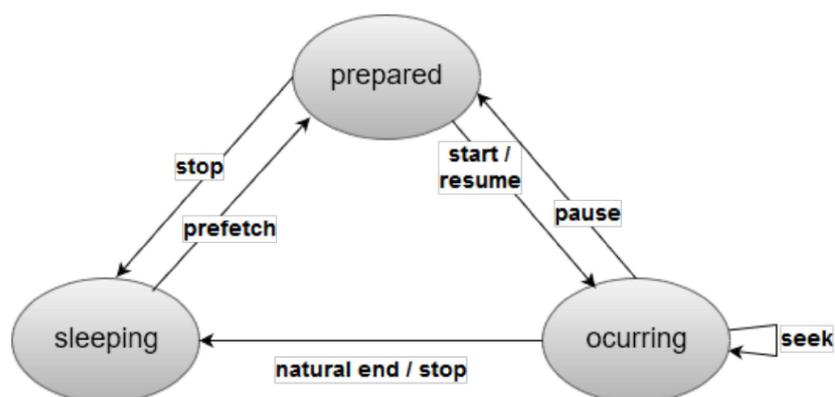
Para uma melhor sincronização de execução, as mídias e os objetos, podem ser controlados por condicionais que definem relações e ações entre objetos e eventos, ligadas por elos, como:

- `onEnd` — O objeto é chamado ao final de outro;
- `onBegin` — O objeto é chamado ao início de outro
- `onSelection` — O objeto é chamado ao ser selecionado em uma área determinada;
- `onPause` — O objeto é chamado quando outro é pausado;
- `onResume` — O objeto é chamado quando outro volta a ser reproduzido;

- `onChange` — O objeto é chamado quando há alterações em alguma propriedade;
- `onRecognition` — O objeto é chamado quando a entrada for reconhecida;
- `onInstant` — O objeto é chamado em um instante temporal (segundos) pre determinado;

STorML possui uma máquina de estados para o evento de apresentação bem similar a de NCL, mas diferindo na remoção do `paused`, e criação de um novo estado, o `prepared`. A máquina de estados, também começa com os objetos em espera (`sleeping`), contudo, momentos antes de uma possível reprodução, é realizada uma preparação (`prefetch`) do objeto, deixando o mesmo preparado (`prepared`) para reprodução (`occurring`). Os estados e transições podem ser observados na Figura 15. A ação de pausa é permitida apenas para a Máquina de Estados de uma Trilha, não sendo aplicável a Cenas.

Figura 15 - Máquina de estados de STorML.



Fonte: Freesz Junior (2017)(18).

Além dos controles convencionais de mídia como `pause`, `resume`, `start` e `stop`, a linguagem também possui a capacidade de saltar a apresentação entre as mídias de uma trilha, ação esta denominada `seek`. Por meio de `seek` é possível avançar (`next`) para próxima mídia da trilha, voltar para a anterior (`prev`), ir para a primeira (`first`), ir para a última (`last`), ou saltar para um determinado tempo específico na `timeline` da trilha.

É possível também efetuar alterações nas propriedades dos objetos em tempo de execução, graças ao elemento `<set>`. Por ele é possível mudar, por exemplo, o atraso de reprodução de alguma mídia, ou até mesmo o posicionamento na tela.

As mídias que compõem as trilhas são definidas pelo formato *MIME Type* e seguindo o padrão Screen-Media Formats definidos pela associação comercial internacional POPAI (Point of Purchase Advertising International) (45). O elemento possui atributos que permitem a personalização da exibição, sendo os principais:

- `dur` — Duração explícita da mídia. Se não definido, é considerada a duração implícita da mídia, trilha ou cena;
- `delay` — Atraso para o início da mídia;
- `opacity` — Define a transparência da mídia;
- `rotate` — Rotação da mídia (-180deg à 180deg);
- `left & top` — Posicionamento (a partir da esquerda superior do visor de exibição);
- `width & height` — Largura e Altura;
- `clipBegin & clipEnd` — Cortes frios nas mídias para selecionar uma parte da mídia a ser reproduzida.

Além das mídias convencionais, há suporte e módulos para geração e exibição de QR Codes e outras mídias chamadas sintéticas, como efeitos sensoriais e vozes sintetizadas por texto.

A linguagem permite entradas de dados de usuários interativamente, como texto, toque, movimento, entre outros. Para realizar tais ações, utiliza-se a trilha interativa e seu objeto `<inputPatternBase>`. A trilha interativa consegue receber dois tipos de padrões de entrada, os simples (`<inputPattern>`) ou compostos (`<compoundInputPattern>`).

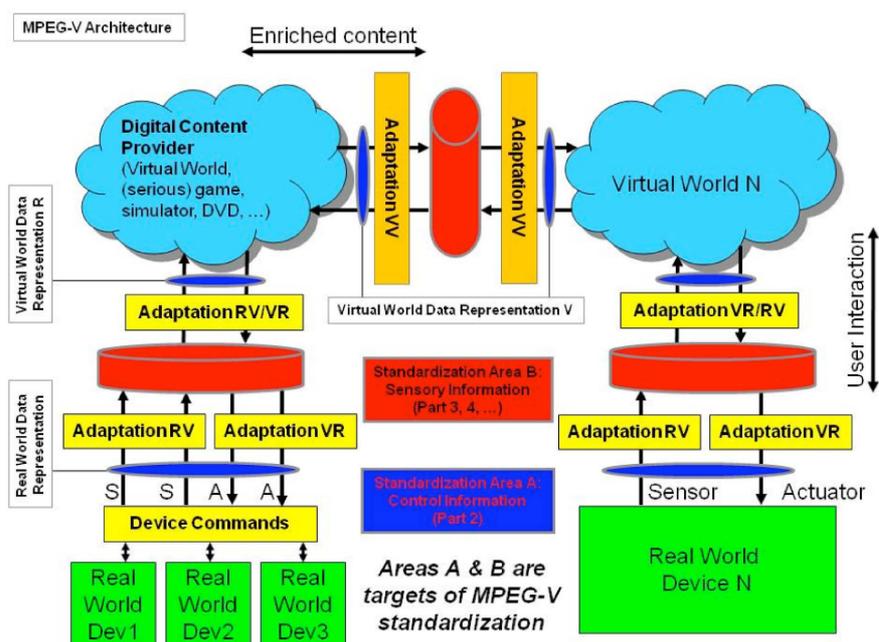
De padrão, STorML tem suporte a três tipos de entradas interativas não convencionais, Gesto, Fala e Escrita, de forma Simples ou Composta. No formato de obtenção simples, permite que um padrão por vez seja obtido. Já no composto, permite múltiplas entradas, baseando-se no trabalho de Nigay e Coutaz (42), onde o autor define que os tipos de entrada composta podem ser “*redundant*”, “*complementary*” ou “*sequentiallyComplementary*”, como já mencionado anteriormente. Em STorML, Friesz (18) define que o redundante (*redundant*) permite que apenas um dos definidos dentro do `<compoundInputPattern>` ocorra. O complementar (*complementary*) necessita que todos os internos ocorram, mas não define ordem, nem tempo. Já o sequencialmente complementar (*sequentiallyComplementary*) necessita que todos ocorram e na ordem definida.

3.7 MPEG-V

O MPEG-V, ou ISO/IEC 23005, é um padrão para uma arquitetura e especificações de representação de informação, com intuito de prover interoperabilidade entre sistemas digitais (mundo virtual) e o mundo real, conforme descrito por Timmerer et al. (58). O padrão é dividido em sete partes na ordem, *Architecture*, *Control Information*, *Sensory Information*, *Virtual World Object Characteristics*, *Data Formats for Interaction*, *Common Types and Tools* e *Reference Software and Conformance*.

A primeira parte do padrão MPEG-V descreve sua arquitetura de forma geral e como ocorre a troca de informações entre dispositivos, sendo as trocas do mundo real para virtual (RV), e do virtual para real (VR). No modelo, toda informação obtida

Figura 16 — Visão geral da arquitetura do MPEG-V.



Fonte: Timmerer, C. (2009) (58)

deve ser processada por um dispositivo de processamento de mídia, que transforma informações obtidas por sensores em representações em dispositivos do mundo virtual, e as geradas no mundo virtual em dados para que atuadores reproduzam os efeitos sensoriais ou mídias no mundo real. A Figura 16 exemplifica a arquitetura do MPEG-V e o processo de comunicação e representação mundo real e virtual.

Em uma exemplificação, um controlador fica responsável por compreender as capacidades de dispositivos do mundo real e como controlá-los. Também gerencia preferências de usuários, e interpreta e transmite informações do mundo real para o virtual, e vice-versa. Por sua vez, “Informação de Sensoriamento” adapta-se para exibir uma mídia ou alguma representação digital em um dispositivo específico, como um vídeo em uma TV.

A segunda parte do padrão (ISO/IEC 23005-2:2018) define a interoperabilidade dos sensores e atuadores, e o funcionamento dos mesmos, em relação à informação captada por sensores no mundo real, e levadas para o mundo virtual (RV), e vice-versa com os atuadores do virtual para o real (VR). Para isso, o modelo descreve ferramentas, exemplificadas em XML com vocabulários para manipulação e interação entre sensores, atuadores e efeitos. Essas ferramentas são referentes a:

- Control Information Description Language (CIDL), definida para fornecer uma estrutura básica da descrição das informações de controle;
- Device Capability Description Vocabulary (DCDV), definido para fornecer uma interface para descrever a capacidade de cada dispositivo sensorial individualmente;

- Sensor Capability Description Vocabulary (SCDV), definido para fornecer uma interface para descrever a capacidade de cada sensor individualmente;
- Sensory Effect Preference Vocabulary (SEPV), definido para fornecer uma interface para descrever a preferência do usuário individual no efeito sensorial específico.

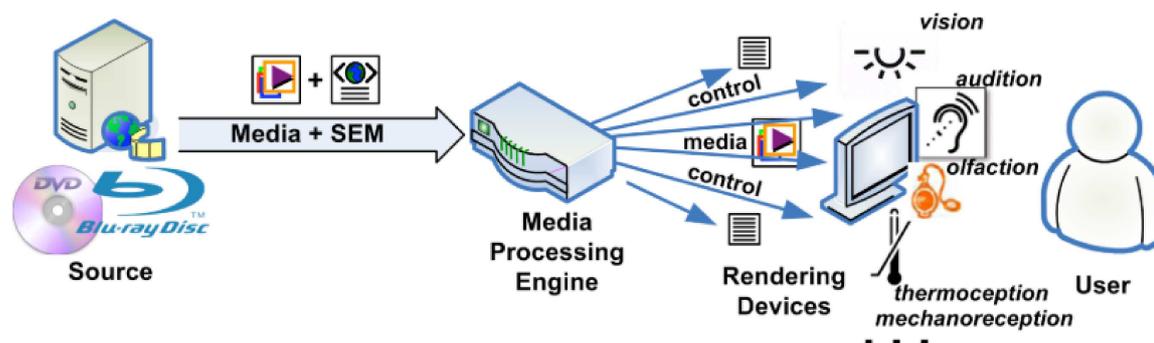
A lista a seguir enumera os atuais descritores de sensores contidos na CIDL, conforme a norma ISO/IEC 23005-2:2018:

- | | | |
|-----------------------------------|--------------------------------|--------------------------------------|
| • Ambient Noise Sensor; | • Mass Air Flow Sensor; | • Altitude Sensor; |
| • Humidity Sensor; | • Light Sensor; | • Distance Travelled Sensor; |
| • Distancy Sensor; | • Angular Acceleration Sensor; | • Array Camera Sensor; |
| • Atmospheric Pressure Sensor; | • Force Sensor; | • Gaze Tracking Sensor; |
| • Position Sensor; | • Torque Sensor; | • Global Position Sensor; |
| • Multi Interaction Point Sensor; | • Gas Sensor; | • Tire Pressure Monitor Sensor; |
| • Body Weight Sensor; | • Camera Sensor; | • Radar Sensor; |
| • Fuel level sensor Sensor; | • Vehicle Speed | • Engine oil temperature Sensor; |
| • Temperature Sensor; | • Microphone Sensor; | • Intake air temperature Sensor; |
| • Velocity Sensor; | • Pressure Sensor; | • Engine RPM Sensor; |
| • Acceleration Sensor; | • Bending Sensor; | • Manifold absolute pressure Sensor; |
| • Orientation Sensor; | • Weather Sensor; | • Intake air temperature Sensor. |
| • Angular Velocity Sensor; | • Speed Sensor; | |
| • Dust Sensor; | • Electronic Nose Sensor; | |
| • Proximity Sensor; | • Motion Sensor; | |
| | • Intelligent Camera Sensor; | |

Para a definição de mídias não convencionais, chamadas efeitos sensoriais (como vento, tato, e afins), na terceira parte do MPEG-V, define-se uma “Linguagem de descrição de efeito sensorial” (Sensory Effect Description Language - SEDL) baseada também em XML. Os metadados com a descrição do efeito são denominados “Metadados de efeito sensorial” (SEM), que podem ser associados a uma mídia convencional (vídeo, som. . .), permitindo a um *player* habilitado (e com dispositivos conectados) reproduzir o efeito sensorial nos demais dispositivos compatíveis com o

efeito. O conceito empregado pelo MPEG-V de como representar e reproduzir os efeitos sensoriais podem ser observados na Figura 17.

Figura 17 — Representação do Conceito da SEDL do MPEG-V.



Fonte: Timmerer, C. (2009) (58)

O modelo prevê duas formas de representação da informação: descritores técnicos, com definições específicas relacionadas a física ou química da representação; ou valores abstratos que enquadram um grupo de ações. Além disso, o MPEG-V permite que, baseado em outras mídias, efeitos sensoriais sejam gerados automaticamente, como associar uma iluminação a um conjunto de *frames* de um vídeo, para que a mesma seja gerada sem configuração humana. Em um nível mais abaixo é possível definir conjuntos de efeitos e parâmetros, para serem utilizados e reutilizados em diversos momentos.

É possível definir marcações temporais para um grupo de efeitos, com intuito de sincronia dos efeitos com mídias contínuas convencionais ou preparação do efeito sensorial. Cada efeito pode ser definido individualmente, com o uso das marcações, tendo suas próprias definições ou utilizando a geração automática, sem que uma mídia seja de fato reproduzida.

No padrão, são definidas propriedades opcionais para organização da reprodução, como:

- Se um efeito deverá ser reproduzido (*activate*);
- a duração (*duration*);
- transição de entrada (*fade-in*) e saída (*fade-out*);
- efeitos alternativos (*alt*) para caso não seja possível reproduzir um efeito específico;
- a intensidade em relação às configurações já predefinidas;
- posicionamento tridimensional para auxiliar na experiência induzida;

- geração automática para gerar automaticamente um efeito específico.

Como citado anteriormente, para realizar efeitos sensoriais de forma mais natural ao homem, o modelo possui uma abstração linguística mais elevada, um “Vocabulário de Efeitos Sensoriais” (SEV). Nele, os efeitos sensoriais desejados pelo programador são escritos com nomes relacionados diretamente com o efeito, como *light* (luz), *colored light* (luz colorida), *Temperature* (temperatura), *Vibration* (Vibração), entre outros. Todas podendo ser configuradas com graus de intensidade ou meio de ação baseadas em escalas tradicionais, como Graus Celsius (temperatura) ou Escala Richter (vibração).

A lista a seguir enumera os atuais efeitos sensoriais descritos pela SEDL, conforme ISO/IEC 23005-3:2019:

- | | | |
|----------------|-------------------------------|-----------------------------|
| • Light; | • Scent; | • Active kinesthetic force; |
| • Flash; | • Fog; | • Tactile effect; |
| • Temperature; | • Color correction; | • Parameterized Tactile; |
| • Wind; | • Rigid body motion; | • Bubble; |
| • Vibration; | • Passive kinesthetic motion; | • Arrayed Light. |
| • Spraying; | | |

A quarta parte prevê a representação dos objetos no mundo virtual, ou seja, como objetos virtuais são reproduzidos em ambientes virtuais, tal qual um cubo em movimento dentro de um monitor. Assim, é possível gerar sensações de interação do mundo real, com um objeto no mundo virtual. Ela é denominada *Virtual Environments*.

A quinta parte descreve a sintaxe e a semântica para a interação e interoperabilidade entre os dispositivos sensores. Para tal, foram desenvolvidas três ferramentas:

- Interaction Interface Description Language (IIDL), define a estrutura básica da interface (ou mensagem) entre um mundo virtual e dispositivos (mundo real) usando um esquema XML;
- Device Command Vocabulary (DCV), define uma interface para dispositivos de comando (atuadores) como parte da IIDL;
- Sensed Information Vocabulary (SIV), define uma interface para aceitar informações fornecidas por sensores como parte da IIDL.

A lista a seguir enumera os sensores cobertos pela especificação de IIDL, conforme ISO/IEC 23005-5:2019:

- Acceleration
- Altitude sensor
- Ambient noise
- Angular acceleration
- Angular velocity
- Array camera
- Atmospheric pressure
- Bend
- Bio
- Blood oxygen
- Blood pressure
- Blood sugar
- Blood type
- Body fat
- Body height
- Body temperature
- Body weight
- Camera
- Center of Mass
- Color camera
- Depth camera
- Distance
- Distance traveled
- Dust
- E-Nose
- ECG
- EEG
- Electrograph
- EMG
- Engine oil temperature
- EngineRPM
- EOG
- Facial expression
- Facial expression characteristics
- Facial morphology
- Force
- Fuel level
- Gas
- Gaze tracking
- Geomagnetic
- Global position
- GSR
- Heart rate
- Humidity
- Intake air temperature
- Intelligent camera
- Light
- Manifold absolute pressure
- Mass air flow
- Microphone
- Motion
- Multi interaction point
- Orientation
- Percentage
- Position
- Pressure
- Proximity
- RADAR
- Spectrum camera
- Speed
- Stereo camera
- Switch
- Temperature
- Thermographic camera
- Tire pressure monitor system
- Torque
- Types
- Vehicle speed
- Velocity
- Weather
- Wind

Na sexta parte, é definida a sintaxe e semântica dos dados utilizados nas demais partes. Por fim, a sétima e última parte descreve testes e formas de validação para verificação de uma aplicação em conformidade ao MPEG-V.

3.8 MODELO SIMM

O modelo SIMM, Simple Interactive Multimedia Model (33), é um modelo que utiliza a representação de sincronização temporal baseada em eventos síncronos ou assíncronos, com um conjunto de entidades simplificadas, se comparado a modelos similares multimídia tal qual o NCM. O modelo visa permitir e facilitar a autoria de documentos hipermídia para usuários com nenhum conhecimento em linguagens de programação ou em modelos de autoria multimídia, evitando as limitações do paradigma de autoria baseado no eixo do tempo, mais conhecido como *timeline*.

O SIMM foi elaborado para o desenvolvimento de ferramentas de autoria multimídia, com uma abordagem simples e capaz de representar aplicações hipermídia, facilitando o mapeamento de aplicações para diferentes linguagens de autoria, tais como NCL e HTML5.

O modelo estabelece a representação de cadeias temporais, itens de mídia e elos síncronos e assíncronos. Os elos síncronos são baseados nas relações temporais de Allen (2), estendidas com a definição de tempo para atraso no disparo de ações.

A visão temporal de documentos produzida pelo SIMM é formada por uma ou mais cadeias temporais, que, são compostas por mídias e elos entre si. As relações são definidas através de uma mídia denominada *mestre*, que serve de referência para que mídias escravas sofram alterações em seus estados de eventos de apresentação, logo após a condição do elo associado ser satisfeita. Para exemplificar, uma ação de *start* entre mídias funciona da seguinte forma. Uma mídia mestra é definida, sendo ela a primeira a ser executada. Posterior a ela, demais mídias escravas são definidas, junto a uma relação entre um tempo t da mídia mestra, a qual mídia escrava utilizará para definir o seu início, *start*.

A tabela 1 produzida por Mattos (33) representa relações possíveis entre mídia mestra e mídias escravas. Na coluna “Símbolo” é apresentado em primeiro na cor verde a mídia mestra, e as demais em cinza as mídias escravas. Assim, é possível visualizar seus respectivos relacionamentos e como ocorrem, baseando-se no tempo da mídia mestra.

O modelo SIMM também é composto por entidades responsáveis por representar características de apresentação de uma mídia, tais como posição, tamanho, e propriedades específicas de mídia. Além disso, é possível uma sincronização espacial relativa a outra mídia, tais como distribuição e alinhamento, seguindo o mesmo conceito de *mestre* e *escravo*. Ou seja, uma vez definida a mídia mestra, a mídia escrava pode se posicionar em relação à mestra, como, por exemplo, um alinhamento baseado na parte do topo, onde a escrava terá seu posicionamento vertical iniciado junto a mídia mestra. A tabela 2 de Mattos (33) exemplifica as relações de aninhamento entre mestra e escrava.

3.9 MULTISEM

MultiSEM, *Multimedia Sensory Effect Model*, estende o modelo SIMM apresentado em 3.8, para suportar a definição e a integração de múltiplos efeitos sensoriais para aplicações multimídia de fim interativo (32). O modelo foi produzido para

Tabela 1 – Relações Temporais do Modelo SIMM

Relação	Símbolo	Descrição
<i>Starts</i>		Escravas iniciam quando o mestre começa
<i>Starts</i> <i>_Delay</i>		Escravas iniciam com atraso quando o mestre começa
<i>Finishes</i>		Escravas terminam quando o mestre finaliza
<i>Finishes</i> <i>_Delay</i>		Escravas terminam com atraso quando o mestre finaliza
<i>Meet</i>		Escravas iniciam quando o mestre termina
<i>Meets</i> <i>_Delay</i>		Escravas iniciam com atraso quando o mestre termina
<i>Met_By</i>		Escravas terminam quando o mestre inicia
<i>Met_By</i> <i>_Delay</i>		Escravas terminam com atraso quando o mestre inicia
<i>Before</i>		Apresenta as escravas sequencialmente, com atraso entre elas, após o mestre terminar

Fonte: Extraída da obra de Mattos(2016)(33)

Tabela 2 – Alinhamento do Modelo SIMM

Relação	Símbolo	Descrição
<i>Top</i>		Escravas são alinhadas pelo topo do mestre
<i>Bottom</i>		Escravas são alinhadas pela base do mestre
<i>Left</i>		Escravas são alinhadas pela borda esquerda do mestre
<i>Right</i>		Escravas são alinhadas pela borda direita do mestre
<i>Center</i>		Escravas são alinhadas horizontalmente pelo centro do mestre
<i>Middle</i>		Escravas são alinhadas verticalmente pelo centro do mestre
<i>Equal</i>		Escravas passam a ocupar a mesma área do mestre

Fonte: Extraída da obra de Mattos(2016)(33)

que profissionais sem conhecimento de linguagens e modelos mulsemídia consigam criar aplicações mulsemídia interativas.

O modelo permite integrar e sincronizar efeitos sensoriais com mídias tradicionais, tanto no tempo, como no espaço, através da modelagem dos efeitos sensoriais, tratadas como nós (*Node*) de um documento, tal qual as mídias tradicionais. A classe *Node* foi adicionada, para representar de forma genérica a mídia (*Media*) e efeitos

sensoriais (*SensoryEffect*), com atributos em comum herdados, como nome do nó, variável de interatividade, tipo, duração, início e fim.

Não são definidas âncoras de conteúdo, sendo essas comumente utilizadas para sincronização de mídias. Em MultiSEM, a sincronização é concretizada por relações temporais entre trechos de tempo. Tais relações são definidas seguindo lógicas como “inicie a mídia A, X segundos após o início da mídia B”, ou “finalize a mídia A no instante Y da mídia B”.

Mídias e Efeitos Sensoriais possuem classes independentes com atributos próprios, além das herdadas da classe Nó. As mídias convencionais possuem o atributo *mediaPresentationProperty*, responsável por armazenar as propriedades já convencionais das mídias. Para efeitos sensoriais, foram adicionados os atributos específicos baseados nas necessidades da classe. O atributo *mediaPresentationProperty*, tal qual o similar na classe mídia, é elaborado para modelar as características de apresentação e renderização do efeito. Cada efeito demanda características próprias, como luminosidade ou posicionamento no espaço. O *autoExtraction* permite que efeitos sensoriais sejam gerados automaticamente de uma mídia de vídeo e áudio, como luz baseada em cores de vídeo, ou vibração sobre os graves do áudio. O atributo *alt* permite definir um efeito alternativo caso o *player* não consiga apresentar o efeito definido. Por fim, é possível definir a prioridade de execução com o atributo *priority*.

Para efeitos sensoriais, o modelo já define os atuais efeitos suportados baseados no padrão MPEG-V, sendo estes: *Wind, Fog, Light, ColoredLight, Flashlight, Scent, Water Sprayer, Impact, Olfactive, Tactile, Temperature, RigidBodyMotion, Vibration, Color Correction, PassiveKinesthetic Motion, Passive Kinesthetic Force-Active Kinesthetic*. Também definem tipos próprios, *Rain, Snow* e *Bubble*.

Variáveis globais foram definidas com o intuito de serem utilizadas em testes de condicionais para relações temporais causais, capazes de serem modificadas por atributos na classe de relacionamento (*relation*). *Relation* permite que as variáveis passem por condicionais simples ou compostas, sendo elas testadas em algum momento predefinido, baseado em eventos de outro nó (síncrono), ou de forma assíncrona por interações.

A classe *SpatialTemporalView* possui o atributo *globalVariableList*, sendo este uma lista de tais variáveis. Essa entidade representa a visão espaço temporal, com relações espaciais, cadeias, e relações temporais. As entidades *TemporalChain* e *TemporalRelation* foram atualizadas em relação ao modelo SIMM para que seus atributos passem a se referir à classe genérica *Node*, permitindo atribuição para mídias e efeitos sensoriais, mas mantendo os relacionamentos temporais de "Mestre" e "Escravo" definidos no modelo SIMM (33).

3.10 COMPARATIVO DOS TRABALHOS RELACIONADOS NO ÂMBITO DE CONTAÇÃO PERVASIVA DE HISTÓRIAS

Como forma de sistematizar os trabalhos relacionados discutidos neste capítulo e melhor identificar as oportunidades de avanços para a modelagem da contação de histórias de forma pervasiva, a Tabela 3, relaciona cada trabalho relacionado com

sua capacidade ou não de endereçar cada requisito levantado na Seção **2.3.1**. Para os casos em que algum tipo de suporte a um requisito seja possível, a tabela informa a funcionalidade oferecida pelo modelo ou linguagem.

Tabela 3 – Resumo comparativo dos Trabalhos Relacionados no âmbito de contação pervasiva de histórias

Resumo dos Trabalhos Relacionados												
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
	Abstração de Alto Nível	Interface uniforme de dispositivos/IoT	Dispositivos do usuário e de terceiros	Adaptabilidade da apresentação, perfil usuário	Variabilidades de ambientes pervasivos	Apresentação por circunstâncias do contexto	Contexto inclui contexto da história	Rastreamento de histórico	Acesso a histórico de outras histórias	Estruturação de experiências e episódios	Experiências c/ múltiplos usuários	Separação entre orquestração e dispositivos
HTML5 (62)	Sim	Apenas Multimídia, demais através de WebSockets e JS	Apenas embutido ou controlado por aplicações cliente servidor com uso de scripts	Sim, através de Javascript e CSS. Perfil a cargo de implementação pelo lado servidor	Sim, através de Javascript	Sim, através de Javascript	Sim, através de Javascript	Sim, através de Javascript e a cargo de implementação pelo lado servidor	Sim, através de Javascript e a cargo de implementação pelo lado servidor	Sim, através de Javascript e a cargo de implementação pelo lado servidor	Sim, através de Javascript e a cargo de implementação pelo lado servidor	Não
NCM/NCL (54,55)	Sim	Apenas Multimídia	Apenas os Embutidos ou na rede local	Sim, <switch> Não há perfil	Não	Sim, <rule>, variáveis, <valueAssessment>	Sim, ocorrências e variáveis da história	Sim, ocorrências e variáveis, persistência local	Não	Sim, <Context>	Sim, Classes de dispositivos	Indefinido
NCL Multimodal/usuário (7,19,20,21)	Idem a NCM/NCL	Melhorias para dispositivos de entrada em relação ao NCM/NCL	Idem a NCM/NCL	Capaz de definir Perfil	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Sim, Idem ao NCL acrescido de perfil de usuário	Idem a NCM/NCL
NCL Efeitos Sensoriais (15,28)	Idem a NCM/NCL	Modificações para contemplar efeitos sensoriais em relação ao NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL	Idem a NCM/NCL
SMIL (65)	Sim	Apenas Multimídia	Dispositivo primário da apresentação	Sim, <switch> Não há perfil	Não	Não	Não	Não	Não	Não	Não	Não
STorM/STorML (18)	Sim	Para entrada <inputPatternBase>. Saída mídias sintéticas em SSML, BML e SEDL	Dispositivos DOOH	Sim, <switch> Não há perfil	Sim	Sim	Não	Não	Não	Não	Não	Não
MPEG-V (58)	Sim	Sim	Sim	Não	Não	Não	Não	Não	Não	Não	Não	Não
SIMM (33)	Sim, Possui ferramenta visual própria (STEVE)	Apenas Multimídia	Apenas os Embutidos ou associado ao Player	Adaptáveis com variáveis e elos	Não	Sim	Sim	Sim	Não	Sim	Não	Não
MULTISEM (32)	Sim, Possui ferramenta visual própria (STEVE 2.0)	Sim, (SEDL)	Apenas os Embutidos ou associado ao Player	Sim, <switch> Não há perfil	Sim	Sim	Sim	Sim	Não	Sim	Não	Não
PST (este trabalho)	Sim	Sim	Sim, distribuídos	Sim, Aura	Sim, Circunstances e estado locked	Sim, Circunstances	Sim, Aura	Sim, Aura	Sim, Aura	Sim, Episodes e Experience	Sim Aura	Sim

4 MODELO PERVASIVE STORYTELLING - PST

O modelo conceitual PST (*Pervasive StoryTelling*), proposto neste trabalho, permite a representação de contações pervasivas de histórias. A contação pervasiva é aquela que necessita de fazer parte do mundo real e do dia a dia em que vive o interator (consumidor da história), capaz de criar uma imersão em informações onipresentes, mas em que as tecnologias empregadas e toda a complexidade de provisionamento tornam-se transparentes. Para isso, parte-se do pressuposto de que sua máquina de apresentação seja implementada sob a forma de um orquestrador ubíquo, presente em todo o lugar e com ciência de todas as informações relevantes para uma dada história e seu interator.

O modelo PST foi desenvolvido com fins de representar, em nível mais alto de abstração, as contações pervasivas de histórias, que recorrem às informações de sensores (*input*) e expõem informações em atuadores (*output*) nativamente e mais próxima do domínio de conhecimento dos criadores de conteúdo. Para isso, o modelo PST se baseou em estudos em colaboração com integrantes do Laboratório de Mídia Digital, da Faculdade de Comunicação da Universidade Federal de Juiz de Fora. Toda a estrutura e nomenclatura do modelo foi elaborada a partir de grupos focais, em que tais pesquisadores se viam livres para discutir a criação das contações de histórias. Sem que os grupos tomassem conhecimento imediato, das discussões foram capturadas terminologias e semânticas recorrentemente por eles usadas nos processos de criação. Assim, o modelo PST busca a assimilação facilitada de suas entidades por criadores de conteúdo, ainda que previamente à concepção de seu mapeamento em uma linguagem visual equivalente, que está fora do escopo deste trabalho.

O modelo PST possui grande similaridade com o ambiente de narrativas criadas para jogos eletrônicos. Além de possuir estruturas e regras de reprodução, também conta com a capacidade de armazenar e utilizar dados do interator, como informações pessoais, informações obtidas por dispositivos associados, e dados gerados por ações do interator durante a reprodução. Tais dados podem ser utilizados pelas entidades de tomada de decisão do modelo.

A modelagem considera também que a contação pervasiva de histórias é inerentemente baseada na ocorrência de eventos assíncronos, dado o mundo aberto em que se desenrola a contação e o conjunto vasto de variabilidades que o caracteriza. Para isso, PST segue um paradigma orientado a eventos. Tais relações se tornam satisfeitas por ocasião de mudanças de estados provocadas passiva ou ativamente, em qualquer parte integrante de uma instância do modelo, desde as mídias convencionais aos interatores, do percurso narrativo seguido ao sensoriamento do ambiente.

Ao contar com um orquestrador ubíquo, as instâncias de PST não definem um dispositivo padrão para a reprodução, mas, sim, dispositivos associados a um perfil de interator, seja porque tal interator os introduziu previamente, seja porque quem os introduziu foi o próprio autor ou provedor do conteúdo. A apresentação de certas partes da história ficará, então, condicionada à disponibilidade dos dispositivos necessários ao longo do período de consumo pelo interator.

Exemplificando, um interator que se interessa por uma contação pervasiva de história poderá iniciar seu consumo se certas circunstâncias necessárias forem satisfeitas, como um conjunto mínimo de dispositivos já associado ao seu perfil. O provedor de tal conteúdo pode associar aos interatores mais dispositivos, tal como uma tela compartilhada em local público. Note que todos esses dispositivos estão apenas associados ao perfil, porém sua disponibilidade certamente variará, como no caso de um celular do interator que pode eventualmente estar fora de área de cobertura da rede ou mesmo sem energia. Conforme mencionado anteriormente, esse tipo de variabilidade é característico de ambientes pervasivos.

Dito isso, este capítulo apresenta o modelo conceitual proposto nesta dissertação denominado *Pervasive StoryTelling* - PST, suas principais entidades, e semânticas. O modelo foi desenvolvido com fins de representar contações de histórias que exploram o uso de sensores e atuadores IoT (inputs e outputs) de forma mais intuitiva a profissionais de criação de conteúdo, com ênfase na reprodução em ambientes pervasivos, com descentralização de “players” e dispositivos de reprodução.

4.1 PERCURSO NARRATIVO

Conforme mencionado no início deste capítulo, o trabalho teve grande apoio nas discussões entre profissionais da comunicação, discussões estas que ajudaram a fornecer os subsídios para a estruturação do modelo. Em reuniões, foi possível capturar a necessidade do modelo em representar as suas entidades de uma forma lógica mais natural aos criadores, mesmo que isso ocorra em detrimento a outras funcionalidades consagradas na literatura da área de hipermídia. Como o modelo é pensado para a criação de contações de histórias, partiu-se da premissa de que as entidades do modelo deveriam, a princípio, auxiliar uma representação visual de percursos possíveis de uma narrativa.

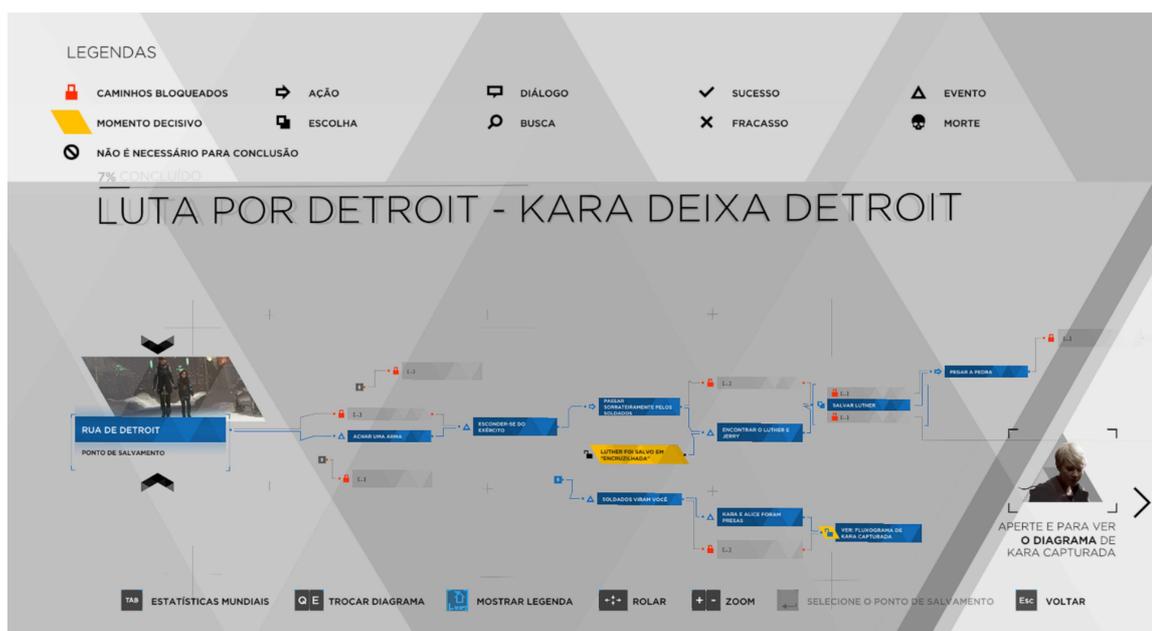
Vale ainda mencionar que, uma vez que o foco é direcionado à construção de contações pervasivas de histórias, tratam-se de narrativas não convencionais, que podem lidar inclusive com elementos atemporais, ou seja, elementos que não possuem horário ou momento específico para ocorrer, mas que impactam diretamente na reprodução (ou não) de outras partes da história. Com isso em mente, vislumbrou-se, com base nos subsídios fornecidos pelos grupos focais do LMD, que o modelo deveria possuir uma semântica de narrativa inerentemente não linear, mas, ainda, capaz de estruturar a ordem, e definir como e quando que eventos devem ocorrer. Tal feito deve ser intuitivo a não programadores, propiciando aos profissionais de autoria a capacidade de compor todos os caminhos que desejassem em uma contação, com suas variações e adaptações ao meio e ao interator.

O modelo permite a construção de “caminhos”, similares a trilhas do NCM em sua versão inicial (37), dedicadas à leitura linear e histórico de navegação. Tais caminhos poderiam ser visualmente perceptíveis em uma possível interface gráfica, em uma representação visual. Eles interligam mídias, efeitos, entradas, e quaisquer outros elementos em seu interior. Eles se aplicam não somente ao processo de autoria, mas podem auxiliar no consumo, já que o modelo permite expressar por quais partes da contação um dado interator passou.

Uma ideia inicial de tal conceito visual que pode ser atingido com a presente proposta pode ser encontrada na obra “Detroit: Become Human” (2018) da Quantic Dream ¹. A história é dividida em capítulos, onde cada personagem tem um pedaço da história a ser contada. Ao abrir um capítulo é possível ver ramificações, mostrando as escolhas do jogador, o impacto delas, e qual caminho (percurso) o jogador seguiu. O jogo também exhibe escolhas antigas que impactaram na liberação de possibilidades atuais, ramificações não seguidas, e detalhes dos caminhos, conforme pode ser constatado na Figura 18.

O modelo PST foi projetado seguindo uma lógica similar, em que seja possível representar uma ramificação de caminhos possíveis em construção por um ator ou disponíveis a um interator, concatenando eventos e informações necessárias para que cada elemento da história seja ativado e reproduzido. Com isso, em uma interface gráfica, o autor conseguirá visualizar uma estrutura de árvore, observando os percursos de sua narrativa.

Figura 18 - Luta por Detroit - Kara Deixa Detroit, Detroit Become Human



Fonte: Retirado de uma campanha do Game: Detroit Become Human (49)

4.2 ENTIDADES ESTRUTURAIIS

PST define um conjunto de entidades estruturais que lhe confere a capacidade de especificar percursos (caminhos) que uma contação de história pode seguir. O modelo apresenta, assim, uma hierarquia em árvore, cascadeando circunstâncias necessárias para que cada objeto seja ativado e reproduzido.

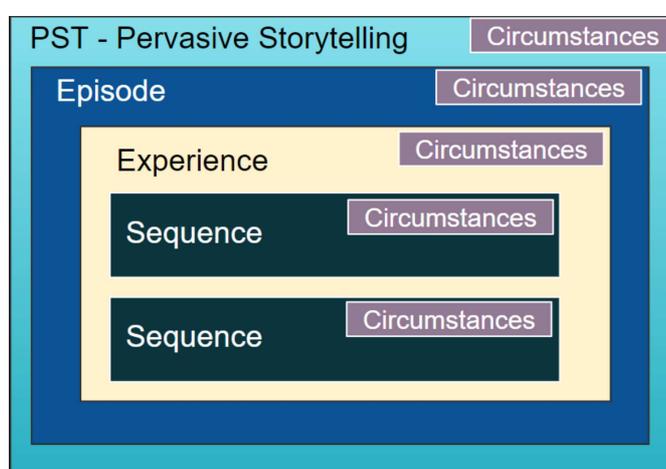
O modelo possui quatro entidades estruturais usadas para dar organização e lógica de apresentação de mídias, efeitos sensoriais e entradas ao longo do

¹ https://en.wikipedia.org/wiki/Detroit:_Become_Human

percurso narrativo: *Pervasive Storytelling*, *Episode*, *Experience*, e *Sequence*, que serão abordados em subseções a seguir. Cada instância dessas entidades tem sua apresentação regida por um conjunto próprio de circunstâncias, definido por meio da entidade *Circumstances*, abordada em detalhes na Subseção 4.2.5.

A Figura 19 ilustra uma visão preliminar do modelo de forma a representar a hierarquia de entidades estruturais. As nomenclaturas e composições foram pensadas para reproduzir no modelo uma lógica de efeito/ação mais intuitiva no contexto de uma contação pervasiva de histórias para profissionais que não sejam da computação.

Figura 19 - Entidades estruturais do Modelo PST agrupadas em visão composicional.



Fonte: Elaborada pelo autor (2021).

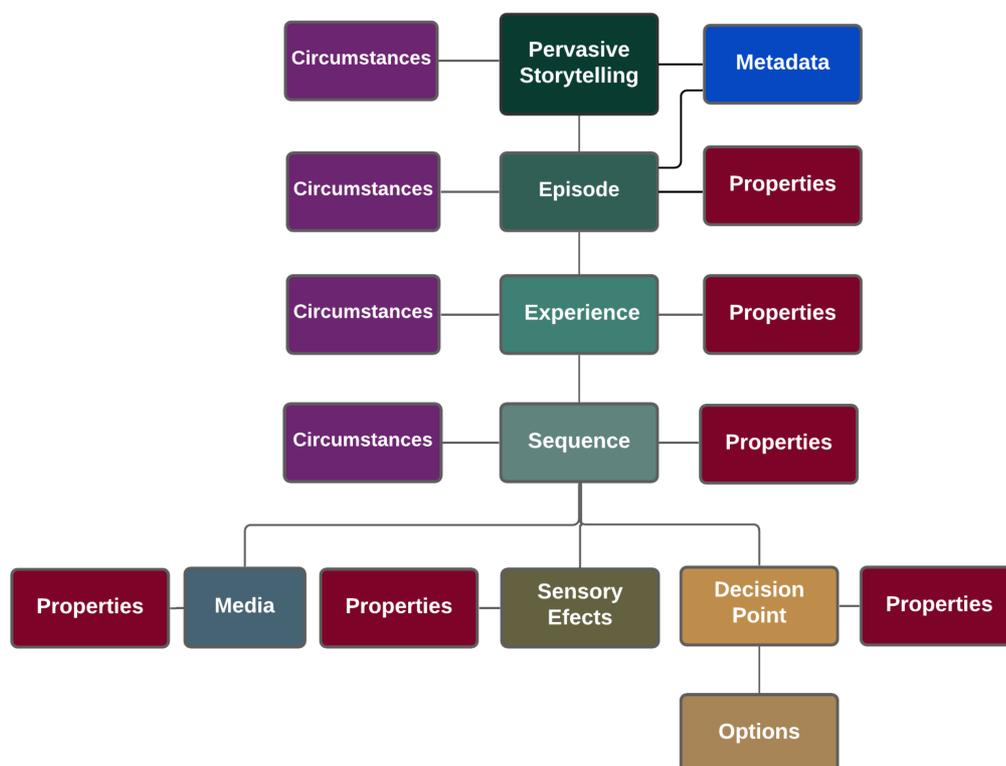
Além da associação a circunstâncias, é esperado que cada entidade estrutural do modelo tenha atributos, como identificador, nome, classe, propriedades e um estado inicial. O identificador, nome e classe podem ser utilizados para que instâncias das entidades possam referenciar-se, umas às outras, ou se sujeitarem a ações em grupo. O estado inicial deve ser definido para que o orquestrador saiba se a instância de uma entidade se encontra apta ou não para a reprodução em um primeiro momento.

As propriedades servem para definir comportamentos mais específicos da instância de uma entidade e de seus descendentes. Quando se define uma propriedade em uma instância superior, todas as instâncias inferiores na hierarquia também carregam tais valores. Os valores podem ser sobrepostos caso haja uma nova definição em uma instância filha. Por exemplo, se declarado em uma instância da entidade PST a propriedade de volume máximo para áudio, todos os sons na contação respeitarão tal valor. Contudo, caso seja declarado um novo valor para volume máximo em uma instância de *Sequence*, todos os sons daquela sequência terão esta última definição como seu valor máximo.

A Figura 20 representa as entidades estruturais rapidamente citadas nesta seção e detalhadas a seguir. A figura inclui, sob a entidade *Sequence*, também as

entidades finais que representam conteúdos de entrada (pontos de decisão) e saída (multimídia, efeitos sensoriais).

Figura 20 - Principais Entidades Estruturais do Modelo PST



Fonte: Elaborada pelo autor (2021).

Um diagrama de classes que ilustra com mais detalhes as entidades estruturais do Modelo PST pode ser encontrado no Apêndice C.

4.2.1 Pervasive Storytelling

A entidade *Pervasive Storytelling* representa o todo da composição de uma contação de história, sendo que ela pode ser composta por uma ou mais instâncias de *Episode*. *Pervasive Storytelling* abriga metadados relacionados à história, como título, gênero, etc. Trata-se da primeira parte, e a base de toda a contação de história, a raiz. Esta entidade é responsável por abrigar toda e qualquer informação relacionada à contação pervasiva. Ela possui propriedades relacionadas à identificação da história, como “id”, nome, classe, e afins, para ser referenciada, modificada ou acessada por terceiros. Além disso, pode incluir metadados importantes para uso pelo orquestrador, por ferramentas de busca de conteúdo ou como forma de introduzir a história antes de iniciada sua contação.

Uma *Pervasive Storytelling* é capaz de importar (*import*) instâncias externas de configurações globais, como estilos, e propriedades padrão de elementos, como mídias, efeitos sensoriais e pontos de decisão.

Dada a possibilidade de se inter-relacionar histórias, o percurso narrativo do interator em uma *Pervasive Storytelling* deve ser mantido em histórico acessível por demais contações a serem consumidas pelo mesmo ou por sua rede de interatores, ressalvadas as necessidades de autorização e preservação de sua privacidade. Isso significa que instâncias das demais entidades estruturais (*Episode*, *Experience* e *Sequence*), por serem partes integrante do percurso, também serão incluídas em histórico. A entidade que permite acesso a tal histórico é a Aura associada a um interator (discutida na Seção 4.3).

Por se tratar da primeira entidade do modelo, é possível definir circunstâncias (condições) para que uma contação seja iniciada para um interator interessado. Assim, é possível aplicar restrições como trava de região por direito de uso sobre a obra, classificação etária, exigir do interator que tenha consumido previamente outra história em específico, não exibição caso possua elementos que ponham em risco a saúde do interator como alergias, foto sensibilidade, ou capazes de afetar alguma síndrome psíquica, entre outras.

4.2.2 Episode

A entidade filha direta de *Pervasive Storytelling* é *Episode*. Um *Episode* possui a semântica de “unidade narrativa”, como em uma série televisiva, na qual cada episódio corresponde a uma porção fechada do todo. O *Episode* também contém metadados de descrição de conteúdo, para a correta identificação ao interator em meio a um catálogo de histórias. Cada episódio é executado individualmente, não sendo previsto, do ponto de vista do modelo, um paralelismo entre episódios, já que o interator seleciona episódio por episódio para seu consumo, ainda que tal ordem possa ser livre, dependendo da obra. Cada episódio possui instâncias de *Experience* a que o interator será exposto.

Em *Episode*, tal qual em *PST*, é possível definir travas de segurança, legais, controle parental e de controle de exibição individualmente para cada episódio. No âmbito de controle de exibição, através da definição de circunstâncias para que ela se reproduzam, os episódios ganham a capacidade de exigir algum consumo mínimo ou perfil desejado pelo lado do interator. Com isso, os episódios podem ser liberados conforme fora o avanço do interator em episódios anteriores da mesma ou outras *PSTs*. Através de dados pessoais públicos do interator, como compras recentes, visitas a lugares importantes para a trama, consumo prévio de alguma outra mídia, entre outros, é possível planejar conteúdo exclusivo para cada perfil de interator.

4.2.3 Experience

Em cada episódio, no que lhe concerne, encontram-se as experiências, representadas como instâncias da entidade *Experience*. Experiências em contações pervasivas podem ser comparadas, de forma análoga, a cenas de um audiovisual, por exemplo. No entanto, se diferem por serem dependentes de circunstâncias assíncronas e por não precisarem ter um tempo de duração pré-definido. As experiências são compostas por instâncias da entidade *Sequence*, que executam em paralelo entre si. A experiência demarca bem ao autor da contação pervasiva as etapas que ele propõe no percurso do interator em um episódio, podendo algumas

delas serem opcionais, inclusive. Assim, o progresso do interator em experiências inseridas em seu histórico possibilita habilitar novos encadeamentos com outras experiências, seja na contação corrente ou em terceiras. No exemplo do jogo Detroit, introduzido no início deste capítulo, a Figura 18 ilustra exatamente o que poderia ser considerado por um autor uma experiência ("Kara deixa Detroit") em meio à contação da história de seu episódio ("Become Human").

Através de uma *Experience*, é possível ao autor proporcionar uma experiência ao interator, que faça sentido em meio à narrativa ser identificada como uma etapa do percurso, que recorre a múltiplos meios de comunicação com um mesmo fim. Logo, um episódio pode possuir mais de uma etapa do percurso, que define a maneira de contar um fragmento da narrativa, a *Experience*, com possibilidades, inclusive, de definições alternativas entre si.

Apesar da similaridade das experiências com as Cenas Adaptativas de STorML, diferente do conceito convencional, em que apenas uma cena pode ser exibida por vez, em um episódio, as experiências são executadas em paralelo, dependendo apenas das circunstâncias que as limitam, permitindo, se fizer sentido para o autor, múltiplas experiências simultâneas. Essa semântica se faz importante.

Além do conceito de experimentação corporal cinestésica e emocional (29), parte da idealização da experiência advém de sua capacidade de propiciar ao autor criar experiências distintas em um mesmo episódio, baseadas em possíveis ações que o interator terá vivenciado, e suas decisões anteriores. Isso permite múltiplas versões de um mesmo episódio, efeito similar ao proposto na obra "Black Mirror: Bandersnatch" (2018) ², da Netflix. Tal efeito é possível devido à capacidade do modelo de capturar, armazenar e disponibilizar dados do interator, por meio do conceito de Aura. Essa capacidade se torna similar à ação de salvamento em jogos, e, neste caso, ser utilizada como componentes lógicos nas circunstâncias e pontos de decisão.

Vale neste ponto salientar que, no cenário pervasivo, a contação de história vai se adaptar àquilo que se mostra disponível no contexto do interator, e, assim, o consumo de um mesmo episódio pode levar a percursos narrativos completamente diferentes, contendo experiências distintas e experiências em comum. Cabe ao autor tal parte árdua da criação em imaginar e produzir tais percursos múltiplos por meio do modelo PST.

Uma experiência pode ser uma referência (*refer*) a outra já existente, para que seja reaproveitada, sem necessidade de recriar uma instância da entidade por completo. Outro atributo importante é a sinalização de repetição, pelo qual pode ser definido quantas vezes a experiência deve ser reproduzida, em repetição. Também deve ser possível sinalizar que, se assim desejar, trata-se de um elemento opcional na reprodução (isOptional). Uma experiência "A" marcada como opcional, determina que o episódio em que está incluída (entidade pai), pode encerrar (fim natural), mesmo que "A" não tenha sua reprodução iniciada.

² <https://www.imdb.com/title/tt9495224/>

4.2.4 Sequence

Cada sequência, criada como instância da entidade *Sequence*, é composta de mídias, efeitos sensoriais e entradas, que são apresentadas uma após o término da outra. *Sequences* são similares à composição *seq* do SMIL, apresentadas em paralelo entre si na mesma experiência, com a diferença de ser possível definir comportamentos distintos de sincronismo entre outras sequências em uma mesma *Experience*. Tal sincronismo ocorre através da especificação opcional de grupos aos quais as sequências são associadas. Esses grupos possuem padrões de sincronismo para que as sequências possam se relacionar. Os padrões de reprodução em grupo definidos no modelo PST são três:

- *syncedSeqGroup*, todas as sequências compartilham de uma mesma *timeline* síncrona;
- *anySingleSeqGroup*, similar ao *excl* do SMIL, onde apenas uma sequência do grupo poderá ser executada em um dado momento;
- *firstOnlySeqGroup*, reprodução excludente onde apenas a primeira sequência a ter circunstâncias satisfeitas deve ser reproduzida.

Apesar dos três modos pré-definidos, o modelo pode ser estendido com novos modos de reprodução em grupo em pesquisas futuras, bastando seu suporte em um orquestrador capaz de implementar as novas lógicas. Assim, o modelo não precisa se limitar a alguns poucos padrões de reprodução de sequências em grupo, aumentando seu leque de possibilidades e experiências possíveis.

Cada sequência possui a duração definida implicitamente pela duração total de todos os seus objetos internos, caso todos os seus objetos tenham uma duração definida, ainda que implicitamente. Quando uma mídia, efeito sensorial ou ponto de decisão (que compõem uma sequência) não possuir um tempo implícito (como um texto a ser lido, um efeito de fumaça ou pergunta interativa), sua duração deverá ser definida explicitamente, ou possuir circunstâncias para reger seu encerramento. Contudo, ainda assim é possível definir de forma explícita uma duração máxima de uma sequência, limitando facilmente o seu tempo para encerramento.

Para as sequências também é possível definir outros atributos, como referência, repetição e sinalização de opcional, tal qual em experiências. Além desses, também é possível definir *delay*, ou seja, tempo em que deve esperar para a sequência iniciar após liberada para ocorrer; os tipos preferenciais de dispositivos em que as reproduções devem ocorrer (*deviceType*), como, por exemplo, definir que efeitos de iluminação ocorram sempre que possível em dispositivos de iluminação RGB.

4.2.5 Circunstâncias

As circunstâncias, instâncias da entidade *Circumstance*, são responsáveis por definir o momento de tomadas de ações sobre as entidades estruturais. Para uma compreensão inicial, as entidades estruturais 20 têm seu controle de reprodução baseado em *circunstâncias*, que seriam um conjunto lógico de regras definido pelo autor, que, sendo verdadeiro, garante uma ação sobre a reprodução da entidade

associada. Em um resumo explicativo de sua semântica, pode-se dizer que “se todas as circunstâncias para que esta parte da contação aconteça forem verdadeiras, efetue a ação definida sobre ela”.

As *circunstâncias* podem se basear em informações obtidas de registros como eventos de transição de estados, estado atual de reprodução, variáveis, valores na *Aura* e propriedades do *interator*. Podem também verificar a existência de dispositivos, e utilizar expressões relacionais e operadores lógicos na composição da sentença lógica. Quando uma *circunstância* e uma respectiva ação é definida para um elemento, o orquestrador verifica se a lógica escrita é verdadeira ou falsa. Quando verdadeira, dispara a ação associada.

Por padrão, o modelo não define quais dispositivos podem ser utilizados durante a reprodução, dependendo do orquestrador verificar os dispositivos disponíveis e preferências do usuário. Contudo, é possível para o autor especificar a categoria de dispositivo do/no qual se almeja obter/exibir as informações. Assim, caso haja múltiplos dispositivos capazes de realizar uma mesma ação, o autor tendo explicitado sua preferência, o orquestrador poderá executar no dispositivo preferencial.

Mais detalhes referentes ao funcionamento das *circunstancias* são apresentados na Seção 4.4, com o intuito de propiciar uma melhor compreensão do leitor, devido ao forte elo da entidade com o controle da reprodução.

4.2.6 Entidades de Entrada e Saída

Apesar de conceitos como *Input* para entrada de dados, e *output* para saída de dados, serem bem difundidos na grande área da ciência da computação, profissionais de áreas distintas podem apresentar dificuldade no momento de abstrair tais representações.

Por isso, o modelo opta por manipular os objetos de representação a serem exibidos ou obtidos, e não o dispositivo que realizará a ação. Sucintamente, as entidades de saída (*Output*) consistem de: Mídias(*Media*) e Efeitos Sensoriais (*Sensory Effect*). Já a entrada (*input*) é representada por Pontos de Decisão (*Decision Point*).

Apesar de mídias e efeitos sensoriais serem ambas saídas, e conseguirem tal qual o NCM (55) representar um objeto em uma mídia qualquer, optou-se por distingui-las para uma melhor compreensão e agrupamento. As mídias convencionais já se encontram bem enraizadas e definidas no âmbito do audiovisual, sendo de fácil compreensão por terceiros. Demais estímulos a outros sentidos como paladar, olfato, tato e afins, e até representações não convencionais como iluminação, são tratadas como efeitos sensoriais.

Já os itens de entrada são enquadrados como pontos de decisão para relacionar de forma lúdica as entradas de dados por interação ou por sensoriamento a ações que ocorrem quando os dados são capturados.

Esse trio de entidades não possui circunstâncias, e deve sempre pertencer a uma *Sequência*. Além disso, cada um conta com atributos específicos relacionados a cada tipo de representação.

4.2.6.1 Mídias e Efeitos Sensoriais

Mídias (*Media*) e Efeitos Sensoriais (*SensoryEffect*), tal qual NCM(55), devem ser especializados em subclasses distintas, para melhor definir a interpretação do conteúdo e suas propriedades.

Mídias podem ser mídias de representação convencionais, como imagem, áudio, vídeo, texto, entre outras, incluindo mídias sintéticas, como abordado por STorM (18). Os Efeitos Sensoriais são representações de saídas não convencionais, presentes comumente em representações *mulsemidia*, como luz, cheiro, vibração, temperatura, entre outros definidos pelos padrões MPEG-V parte 2: *Control Information*(23) e parte 3: *Sensory Information* (24).

Essas entidades de saída devem sempre especificar o seu tipo (vídeo, bolha, entre outros), junto a uma referência ao conteúdo no caso de Mídia, e suas respectivas propriedades de reprodução, para que o orquestrador consiga reger sua apresentação.

Como este trabalho não apresenta uma linguagem baseada no modelo, o mesmo não define quais as propriedades obrigatórias de cada mídia ou efeito sensorial, ficando a cargo de trabalhos futuros o estudo e desenvolvimento de uma linguagem, preferencialmente visual.

Para representação de mídias convencionais, já existem inúmeros trabalhos consagrados, capazes de definir propriedades das subclasses, como NCL (54). Para a representação de efeitos sensoriais, já se encontram padrões de representação e comunicação mundo real-digital, como MPEG-V (58), e representações visuais mais abstratas como as utilizadas por STEVE (32).

Entretanto, vale mencionar que uma linguagem não deveria obrigar o autor a especificar manualmente todos os elementos e valores referentes a cada categoria de efeito conforme é descrito na série ISO/IEC 23005, devido a sua grande complexidade. As definições poderiam ser representadas até mesmo de forma mais subjetiva, como “quente” e “frio”. E o conceito de quente e frio poderia ser apontado por propriedades de preferências do interator, ou, na sua ausência, valores padrões e de fácil compreensão. Caberia então ao orquestrador realizar o mapeamento entre formatos de representação de diferentes níveis de abstração.

Assim, um autor com maior domínio poderá criar efeitos precisos, e melhores adaptados conforme a subjetividade definida pelo autor e/ou interator. Isso se faz importante devido à variação de níveis de sensações entre seres humanos. Exemplificando, pessoas morando em regiões diferentes, acostumadas com climas específicos, variam sua sensação de calor e frio, sendo então a definição exata da temperatura em Celsius, uma forma insatisfatória de referenciar a sensação.

Uma entidade de saída sempre deverá ser reproduzida no primeiro dispositivo disponível e possível de reprodução, cabendo ao orquestrador descobrir como representar o efeito ou mídia definida. Veja que não se deve representar um dispositivo, mas sim um efeito ou controle possível de ser reproduzido. Logo, o orquestrador deverá, baseado nas preferências do interator, do autor, e da disponibilidade, buscar reproduzir a mídia ou efeito. O autor deve conseguir definir, se assim desejado, qual a categoria de dispositivo preferencial da reprodução dos efeitos.

Para ser possível um reúso de mídias e efeitos, é possível declarar uma entidade de saída que referencia a outra, através de uma referência a seu identificador. As propriedades e atributos não únicos também devem ser reaproveitados, mas permitindo uma sobrescrita caso seja redeclarado. Além disso, as mídias devem conseguir referenciar por meio de URIs os conteúdos a serem obtidos para reprodução.

4.2.6.2 Ponto de Decisão

Conforme apontado por Guedes et al. (21) uma Interface de Usuário Multimodal (Multimodal User Interface - MUI) consiste na capacidade de se obter *inputs* de estímulos de várias modalidades. Em um ambiente pervasivo com a possível disponibilidade de diversos dispositivos de modalidades distintas, uma interface MUI torna-se bem-vinda e necessária, dado que o modelo não deve contar com um dispositivo padrão de entradas de usuário.

Nos grupos focais junto ao Laboratório de Mídias Digitais (UFJF), evitou-se trazer às discussões os termos da computação, como “entrada” (*input*) e “saída” (*output*), de modo a deixar livre e compreender como profissionais da área de comunicação referenciariam a semântica de exibição e captura de dados/informação em um processo de criação. Notou-se, então, que entradas (*input*) ou informações acessíveis no que viria a ser chamado *Aura*, eram constantemente referenciadas como “escolhas” do interator, explícitas por meio de interação, ou implícitas, por meio de sensoriamento e histórico.

Em certas discussões do grupo focal pôde-se capturar a ideia de um **ponto** no qual o interator **decida** o rumo da história, que começou a ser denominado frequentemente como “Ponto de Decisão”. Nota-se também que ponto de decisão é um termo genérico o suficiente para abrigar tanto a entrada de informação por meio da interação do usuário quanto a entrada por meio de sensores, preferências e histórico.

Um ponto de decisão é uma instância da entidade *DecisionPoint* e estabelece um conjunto de opções possíveis, das quais, caso alguma seja verdadeira, dispara-se uma ação dentro da história sendo contada.

Portanto, o Ponto de Decisão permite que o autor associe lógicas de interações ou avaliação de dados da *Aura* do interator. Sendo assim, todo Ponto de Decisão é composto por um conjunto de Opções (*Options*), em que cada opção contém expressões lógicas que determinam quais dados precisam ser obtidos, quais os valores esperados para que a opção se torne verdadeira, e, nesse caso, qual a ação a ela associada deve ser disparada. É possível usar informações pertinentes a interatores e definir de que perfil se espera a interação. Para evitar falta de respostas, que pode acontecer devido às já comentadas variabilidades do ambiente pervasivo, um ponto de decisão deve especificar o que ocorre caso não haja opções satisfeitas em um limite de tempo definido.

Quando um *Ponto de Decisão* é definido, o autor poderá declarar dois atributos, “*timeout*” e “*timeOutAction*”. Tais atributos existem para resolver casos em que não seja possível obter os dados de entrada do interator, seja por omissão do mesmo,

ou algum problema nos dispositivos de captação. O *Timeout* define o tempo máximo que o orquestrador aguardará, até que uma opção se torne válida. Durante o tempo definido, o orquestrador deverá verificar constantemente se alguma opção se tornou verdadeira. Após o tempo do *timeout* se esgotar, e nenhuma opção for verdadeira, uma ação definida no atributo denominado *timeOutAction*, será executada. As ações de *timeOutAction* possuem a mesma estrutura que as ações (*action*) na entidade *Option* (Opção).

No Ponto de Decisão (DP), encontram-se as opções, instâncias da entidade *Option*, possíveis caminhos a serem seguidos em resposta a interações dos *interatores* ou, passivamente, a partir de dados capturados de sensores ou salvos na *Aura*). Opções formam uma lista de condicionais que disparam respectivas ações a serem tomadas, com uma estrutura similar a controladores de fluxo condicional de linguagem de programação (como em C), chamados de *switch case*. Dentro de cada opção, há um atributo *condition*, similar à propriedade de mesmo nome na entidade *Circumstance*, definindo as condições para que aquela opção seja verdadeira, o que leva à execução da respectiva ação definida no atributo *Action*. Ao ter uma das opções validada, o Ponto de Decisão é encerrado, ou seja, apenas a primeira opção validada como verdadeira tem sua ação disparada.

A condição (*condition*) presente dentro de uma instância de *Option* segue a mesma forma de validação que *Circumstance*, podendo utilizar de operadores lógicos e expressões relacionais. Quando um *Ponto de Decisão* é iniciado, o orquestrador deve buscar compreender quais os *inputs* necessários, acionar os dispositivos capazes de os obter, ou obter valores da *Aura* necessários, verificando-os constantemente, até que uma *Opção* se torne verdadeira.

O modelo não exige que o autor defina explicitamente o dispositivo do qual o objeto de entrada será capturado, mas o seu tipo. A definição se limita ao efeito de entrada ou saída e seus parâmetros, deixando para o orquestrador na nuvem decidir de quais dos dispositivos registrados conseguirá obter os dados solicitados em uma opção.

Caso em uma opção não seja definida uma condição, a opção será dada como verdadeira, e a ação relacionada será disparada automaticamente. Caso haja mais de uma *opção* verdadeira simultaneamente, apenas uma opção é escolhida, não havendo comportamento definido para tal escolha.

É importante mencionar que um conjunto de funcionalidades de uma MUI, definido por Nigay e Coutaz (42), em relação às combinações de interação entre modalidades, torna-se possível de ser representado, na semântica da lógica embutida em *condition* da entidade *Option*. Graças a operadores lógicos utilizados em *condition*, é possível, com o uso do operador *or*, definir o conceito de redundância, *redundant*, quando apenas uma expressão precisa ser verdadeira; com o operador *and*, garante-se que todas as expressões na condição precisam ser verdadeiras, mas sem ordem definida, sendo complementares, *complementary*; por último, usando um operador de grupo, *seq()*, determina-se que as expressões por ele agrupadas devem todas acontecer e na ordem em que foram declaradas, sendo, assim, sequencialmente complementares, *Sequentially Complementary*.

No atributo *action*, por sua vez, é possível definir o que acontecerá caso a

condição seja verdadeira. O modelo não limita quantas ações poderão ser definidas e executadas por vez. Elas podem ser direcionadas a qualquer instância de entidades estruturais ou finais da contação corrente. Podem também modificar valores de propriedades, como os da *Aura* do interator. As ações possíveis são: *start*, que dá início à reprodução de uma instância que esteja habilitada; *pause*, que fará uma pausa em uma instância que esteja ocorrendo; *stop*, para encerrar a apresentação de uma instância que esteja ocorrendo; *resume*, que retomará a apresentação de uma instância pausada; *set*, para armazenar ou modificar valores de propriedades da *Aura* do interator ou de instâncias de entidades quaisquer do modelo *PST*; *seek*, para avançar ou retroceder uma mídia, efeito sensorial ou uma sequência.

Observa-se, ainda, que o orquestrador irá disparar a ação em paralelo, finalizar o Ponto de Decisão e dar prosseguimento à sequência, já que esta pode ter mais mídias/efeitos/pontos de decisão na continuidade. Sendo assim, caso seja do desejo que a sequência vigente do ponto de decisão encerre assim que uma opção se tornar verdadeira, o autor deverá na ação incluir um *stop* na própria sequência. Outra opção é que o Ponto de Decisão seja o último item da sequência.

Vale ressaltar que, apesar da entidade *Circumstance* prover a capacidade de uso como uma condicional para avaliação e tomada de ações, a mesma exige que os recursos desejados para verificação sejam constantemente avaliados, o que ocasionaria em uso de recursos em momentos desnecessários. Além disso, a mesma não propicia eles, dado que define ação sobre o próprio elemento em que foi associada. Os *Pontos de Decisão* garantem então um recurso importante para propiciar gatilhos diretos, e estabelecer um conjunto de ações de livre especificação.

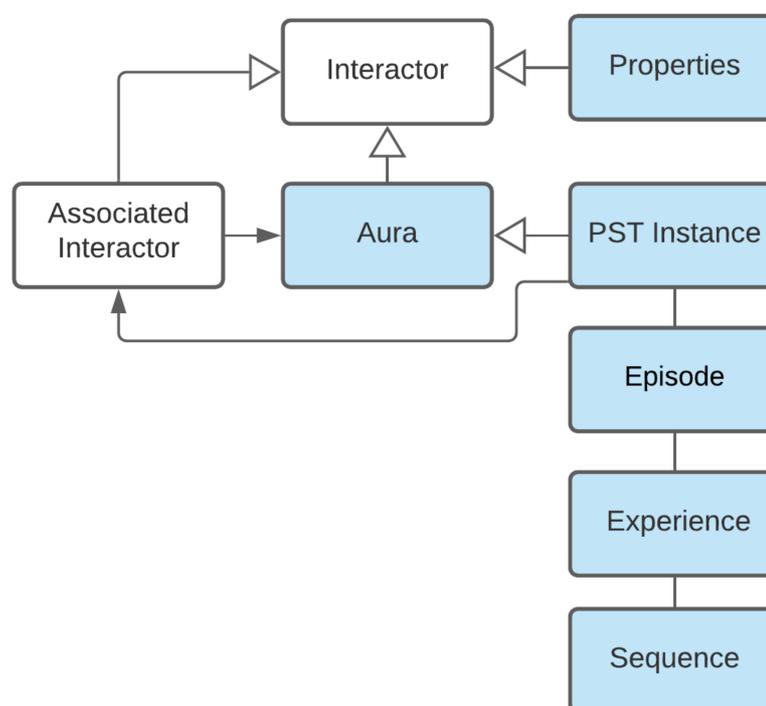
Este modelo não possui por objetivo definir os padrões de comunicação e representação de informações de dispositivos de entrada (*input*), tal como proposto pelo MPEG-V (58) pela ISO/IEC 23005-5:2019. Entretanto, além da especificação técnica de como representar os dados que se deseja serem capturados, os mesmos poderiam possuir representações alternativas mais palatáveis aos profissionais do domínio específico de criação de conteúdo.

4.3 INTERACTOR E AURA

Devido à sua capacidade ativa de consumo e interação, os comumente denominados usuários ou espectadores, neste trabalho, foram nomeados “interatores”, seguindo o conceito de Janet Murray (38). Para ser possível armazenar informações de interatores, como dados pessoais e preferências, que podem impactar na contação da história, é definida a entidade *Interactor*. A relação entre as entidades criadas pertencentes a um interator pode ser observada na Figura 21.

Com grande referência em jogos digitais, nos quais o progresso de cada jogador é armazenado, a entidade *Interactor* possui uma entidade *Aura* associada, que coleta toda a ciência de contexto de suas respectivas instâncias de contações pervasivas nas quais iniciou consumo. A *Aura* possui a capacidade de armazenar dados referentes à contação, similar a jogos com sistema de *Karma*, nos quais as atitudes registradas do jogador servem de embasamento para eventos e tomadas de ações. Seguindo uma lógica similar, o interator carrega consigo quem ele é na

Figura 21 - As Entidades do Interactor e Aura



Fonte: Elaborada pelo autor (2021).

contação, sendo ele resultado de suas escolhas, criando assim, sua *Aura*. Esses dados podem ser utilizados para validações lógicas em *Circumstances* e *Decision Points*, sendo essas as entidades de tomada de decisão. A *Aura* armazena todo o percurso trilhado pelo interactor em uma instância de contação de uma *PST* em que se encontra inscrito, assim como os estados das entidades estruturais definidas em 20.

A entidade *Properties* possui a função de armazenar informações pessoais, preferências do indivíduo e associações a perfis, tais como: necessidades especiais, como deficiências físicas e/ou sensoriais, de modo a permitir a opção de criação de caminhos da contação de história que considerem acessibilidade e inclusão; valores para parametrização de sensações abstratas e individuais, como exemplo a temperatura que corresponde a frio e a calor para o interactor, com o intuito de possibilitar ao autor uma melhor adaptação da experiência de efeitos sensoriais e mídias convencionais; preferências de reprodução como idioma, legendas, velocidade, dispositivos primários, secundários, e afins; dados para controle parental e emocional, para serem considerados para que um objeto seja ou não reproduzido. Tais exemplos são apenas algumas das possibilidades, no entanto, o modelo não limita quais informações podem ou não ser solicitadas, obtidas e armazenadas, cabendo ao autor e ao orquestrador lidarem com permissões e avisos sobre uso de dispositivos e dados, de acordo com as legislações locais.

Com essas informações, os autores terão a capacidade de definir quais cami-

nhos cada interator percorrerá, abrindo e fechando portas para etapas da contação da história, baseando-se em cada tomada de decisão e elementos disponíveis por quem a consome, similares a jogos do gênero RPG, onde toda ação de um jogador é armazenada e impacta diretamente em sua experiência.

4.3.1 Múltiplos Interatores

A entidade *Associated Interactor* permite que dados de interatores sejam cruzados ao longo de contações de histórias, permitindo a participação de múltiplos interatores que se juntam em um círculo social. Exemplificando, durante o consumo de uma história por um interator, caso a plataforma de provisão das histórias tenha suporte a relações de amizade entre interatores, e um interator amigo inicie uma instancia de PST em comum e com suporte a múltiplos interatores, uma contação da história poderá ser consumida em conjunto, registrada individualmente na aura de cada interator, mas cruzando informações e dispositivos de cada participante quando necessário.

Infelizmente, o modelo não descreve inicialmente como distinguir múltiplos usuários locais utilizando de um dispositivo de entrada compartilhado simultaneamente, para salvar o progresso em *auras* distintas. Assim, o progresso será registrado apenas na aura do interator ao qual aquele dispositivo está associado. Contudo, é possível a identificação de quem fez cada entrada quando cada um dos interatores usa seus próprios dispositivos, tal qual ocorre em consoles para jogos digitais e seus múltiplos controles (*joysticks*), através da distinção entre dispositivos de entrada, e da associação dos itens a um interator.

Sendo assim, o modelo permite as modalidades de interação definidas por Guedes et al. (2016) (19):

- “Put-That-There”, podendo haver diferentes modalidades de entrada e / ou saída para um único usuário interagindo com o sistema, e salvando seu percurso na aura.
- “I-Get-That-You-Put-It-There”, permitindo em um interator inicie uma ação, e outro interator associado registrado naquela contação termine aquela ação, através de dispositivos distintos, e mesmo em execuções distintas, desde que ambos estejam sincronizados com a nuvem que coordena todas as apresentações;
- “Anyone-Get-That-Someone-Else-Put-It-There”, em narrativas coletivas interatores inscritos, possam, independentemente de quem seja, iniciar uma ação, e qualquer outro inscrito na mesma contação de história a continue, sendo ela coordenada pelo orquestrador em nuvem.

4.4 MÁQUINA DE ESTADOS DAS ENTIDADES

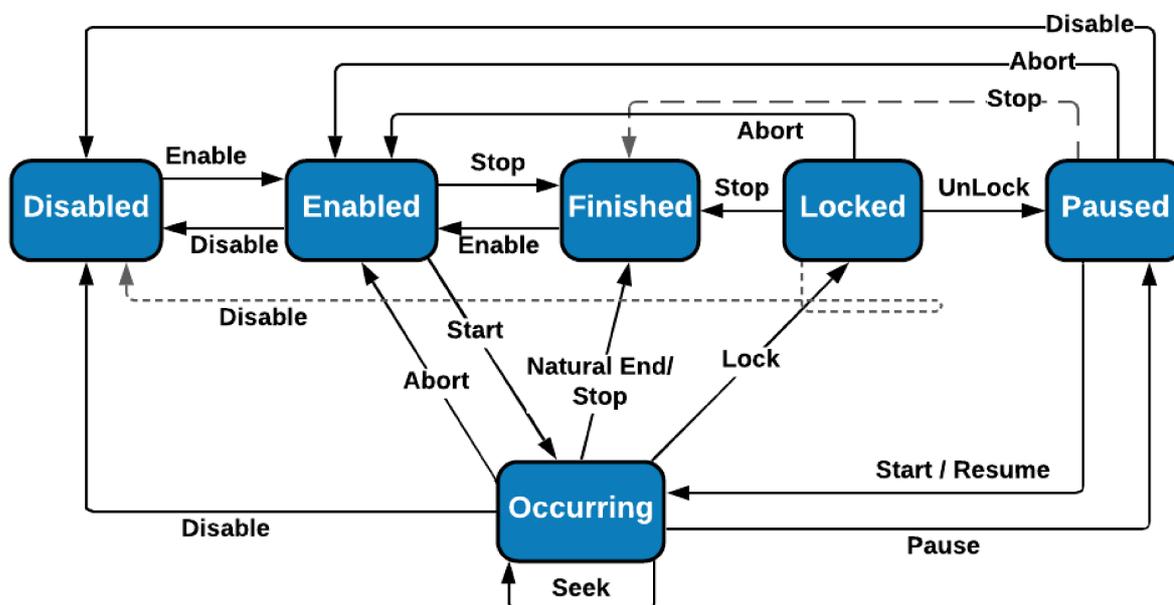
O modelo propõe, junto a sua estrutura, uma máquina de estados associada à apresentação de suas entidades, devido ao seu alto grau de abstração, considerando

a atuação das circunstâncias e das variabilidades que podem levar a uma invalidação de circunstâncias.

Nesta seção, são detalhados o processo de uma reprodução, e os estados e transições que compõem a máquina de estados de uma entidade estrutural do modelo.

A máquina de estados atrelada à apresentação de cada entidade estrutural do modelo PST estende estados comumente encontrados em modelos hipermídia, tal qual NCL (55), como “*Occurring*”, para sinalizar que o elemento (instância de uma entidade) está em execução, e “*Paused*”, sinalizando que o elemento se encontra pausado. O estado “*Enabled*” sinaliza que o elemento está apto a ter circunstâncias avaliadas. Outros estados são definidos para simbolizar que um elemento se encontra incapaz de ser reproduzido, como desabilitado (*Disabled*), e encerrado (*Finished*). Também há um estado para trava automática de reprodução, denominado “*Locked*”, normalmente atingido devido a uma perda de circunstâncias. Transições típicas de sistemas hipermídia também foram reaproveitadas, como *start*, *stop*, *abort*, *pause*, *resume*, *natural end* e *seek*. Contudo, o domínio da contação pervasiva de histórias exige que novas transições sejam suportadas, devido às variabilidades de ambientes pervasivos. Todos os estados e transições possíveis entre eles podem ser visualizados na Figura 22.

Figura 22 - Máquina de Estados da Apresentação.



Fonte: Elaborada pelo autor (2021).

Inicialmente, por padrão, uma instância de uma entidade estrutural tem seu estado como habilitado (*Enabled*). Tal estado sinaliza que tal elemento se encontra apto a ter suas circunstâncias observadas. Conforme mencionado na Seção 4.2.5,

as circunstâncias possuem condicionais que são avaliadas tal qual testes lógicos. Dado como verdadeiro o teste lógico da circunstância de um elemento, a máquina de estados será transicionada para o estado *Occurring* e a reprodução do elemento deverá ser iniciada.

Apesar de o estado *Enabled* ser o padrão de um elemento, o autor tem autonomia para definir outros estados iniciais, como *Disabled*. Caso o elemento se encontre desabilitado (*Disabled*), suas circunstâncias não são avaliadas, por consequência não pode ser reproduzido. Essa característica permite ao autor definir elementos que jamais serão reproduzidos, até que algum evento definido pelo mesmo o habilite, como um *Ponto de Decisão*.

Quando as circunstâncias forem verdadeiras, o elemento é então iniciado, e move-se para o estado de ocorrendo, *Occurring*. O ato de ocorrer, não significa que uma mídia ou efeito sensorial está em reprodução, mas sim que aquele elemento estrutural está ativo, e que, por consequência, seus filhos podem ter suas circunstâncias observadas, se eles estiverem em estado *Enabled*. Qualquer elemento, exceto os da entidade PST, não pode ocorrer ou transicionar de estado sem que o pai esteja ocorrendo. Se o pai deixa de ocorrer, os filhos também deixam, com a mesma transição.

A título de ilustração, o console Xbox 360 da Microsoft ³, possui por padrão um modo de funcionamento que, caso a conexão com um controle seja perdida, ou o jogador de um jogo utilizando o dispositivo Kinect saia da zona de captação do sensor, o jogo é pausado, impedindo o mesmo de retomar até que o controle seja reconectado, ou o jogador volte a ser captado. Todo elemento ocorrendo (*Occurring*) tem mantidas suas *circunstancias* em constante avaliação. Esse mecanismo permite ao orquestrador agir automaticamente caso um elemento que já esteja ocorrendo tenha circunstâncias perdidas, ou seja, se tornem falsas. Nesse caso, deve-se travar a apresentação de tal elemento e de seus descendentes, entrando no estado *Locked*, até que volte a ter suas circunstâncias válidas para reprodução. Esse estado é de uso exclusivo do orquestrador, ou seja, o autor não tem permissão para forçar um objeto a este estado.

Após o destrave (*unlock*), porque circunstâncias voltarem a ser verdadeiras, o estado do elemento transita para *pause*, permitindo ao interator retomar sua reprodução, se e quando desejar, e continue o *loop* de avaliação até que o elemento seja encerrado. O estado *Locked* foi elaborado para garantir um controle mais refinado da reprodução, visto que a perda de conexão de dispositivos em ambientes pervasivos é comum e esperada, além da capacidade inerente da constante variação de informação do meio onde se encontra um interator.

Locked é um estado incomum na literatura de modelos hipermídia convencionais. Em trabalhos que estendem NCL para uso de dispositivos IoT, como de Rodrigues (15) e Rodrigues et al. (13), são propostas metodologias para firmamento de conexão entre dispositivos, e formas de lidar com atrasos e problemas de sincronia, também abordadas por Josué (27). Pouco se discute sobre como uma máquina de apresentação ou formatador deve lidar com a perda dos aparelhos e/ou situações de incapacidade de reprodução.

³ <https://en.wikipedia.org/wiki/Xbox360>

Para garantir que uma entidade estrutural encontre o seu fim natural, quando todos os seus filhos estiverem desabilitados e/ou finalizados, a máquina de estados do elemento em questão transaciona para *Finished*.

Por fim, o último estado possível é *Paused*, no qual um objeto é paralisado, tendo a sua reprodução e de seus descendentes pausadas, ficando no aguardo de uma ação externa, de um ponto de decisão, ou circunstâncias de retomada, se diferenciando da máquina do modelo STorM (18), que não permite que certos elementos sejam pausados, visto que uma pausa na cena acarretaria uma impossibilidade na geração de um evento que pudesse resultar em sua retomada (*resume*). Contudo, cabe ao autor ter consciência sobre tal fato, para que um objeto não fique em pausa infinitamente. Uma sugestão a desenvolvimentos futuros relacionados ao tema é a criação de validadores capazes de avisar ao autor sobre possíveis problemas e inconsistências em sua narrativa, como uma pausa não retornável ou elemento sem capacidade de ser iniciado.

Em suma, os estados de reprodução dos elementos estruturais são:

- *Enabled*, define que o elemento está pronto e esperando o seu momento de ser reproduzido;
- *Disabled*, define que o elemento não poderá ser reproduzido, estando desabilitado (não terá suas circunstâncias observadas, logo, não o reproduz);
- *Occurring*, sinaliza que o elemento se encontra em reprodução;
- *Paused*, o elemento encontra sua reprodução pausada;
- *Locked*, definindo que o elemento atualmente se encontra travado e impossibilitado de reproduzir, por falta de circunstâncias;
- *Finished*, informa que o elemento já foi reproduzido em sua totalidade.

Grande parte das transições foram reaproveitadas da linguagem NCL, como *start*, *stop*, *abort*, *pause*, *resume*, *natural end* e *seek*. Contudo, o modelo exigiu que novas transições fossem criadas, devido a novos estados, além de limitações de quando cada estado é possível transacionar para outro, como variações sobre ação advinda pelo orquestrador, interator ou circunstâncias do autor.

Um elemento pode ser habilitado (*enable*), quando um elemento se encontra em um estado que não possui autorização de ser executado, sendo estes (*Disabled* ou *Finished*), através de ações em pontos de decisão predefinidos pelo autor. Outra forma de transicionar para o estado de *Enabled*, ocorre quando um objeto tem sua reprodução abortada em algum dos seguintes estados: *Occurring*, *Paused*, *Locked*. Quando isso ocorre, o elemento é reiniciado, tendo todo seu progresso perdido. Caso um elemento venha a ser reproduzido novamente, ou seja, transicione de *Finished* para *Enabled*, deve ser criada uma nova instância na aura do interator, sem perder os dados antigos, mas se tornando a atual.

Um elemento habilitado (*Enabled*), pode transicionar para *Occurring* através de uma ação de *start*. Uma vez ocorrendo, caso alguma circunstância do elemento torna-se falsa, ele é travado (transição *lock*), e caso voltem a serem verdadeiras as

circunstâncias, ele é destravado (transição *unlock*), sendo estas ações exclusivas do orquestrador.

Uma vez pausado por ação do interator ou do autor, por meio da transição *pause*, o mesmo pode ser retomado por um *resume*. Após a reprodução por completo, o elemento alcança o seu fim natural, sendo terminado com a transição *natural end*. Caso esteja em algum estado de reprodução distinto de *Finished* e *Disabled*, o mesmo poderá ter um fim forçado através da transição *stop*.

Um elemento pode ter sua reprodução desabilitada, tornando-se inacessível para reprodução (estado *Disabled*), por meio da transição *disable* a partir de qualquer estado, exceto se já tenha encerrado (estado *Finished*).

A ação *seek* permite que avance/retroceda a reprodução para um elemento dentro de uma sequência, ou para uma marca temporal dentro de uma mídia ou de uma sequência em que todos seus elementos tenham duração predefinida. Caso a sequência ou elemento da sequência que sofra um *seek* esteja inscrita em algum grupo de sincronia, o orquestrador deverá refletir a ação nos demais membros do grupo.

A tabela 4 elenca todos os estados, transições possíveis, e para qual estado se é transacionado.

Tabela 4 – Estados e Transições da Máquina de Estados.

Estado primário	Transição	Estado Transicionado
Disabled	Enable	Enabled
Enabled	Disable	Disabled
	Start	Occurring
	Stop	Finished
Finished	Enable	Enabled
Occurring	Abort	Enabled
	Disable	Disabled
	Lock	Locked
	NaturalEnd/Stop	Finished
	Pause	Pause
	Seek	Occurring
Paused	Abort	Enabled
	Disable	Disabled
	Start/Resume	Occurring
	Stop	Finished
Locked	Abort	Enabled
	Lock	locked
	Stop	Finished

Fonte: Elaborada pelo autor (2021).

Para possibilitar maior riqueza ao modelo, e propiciar ao autor que defina momentos exatos de gatilhos de eventos, cada transição deve ser capturada pelo orquestrador e de possível referência pelo autor, permitindo assim que circunstâncias as utilizem e ações sejam tomadas entre mudanças de estados de reprodução, tal qual NCL o faz com os chamados “Papéis de Condição” e “Papéis de ação”.

Devido à similaridade de alguns Papéis de Ação, com ações sobre os estados de reprodução deste modelo, foram reaproveitados alguns dos conceitos e definições de NCL, acrescidos das transições criadas neste modelo. Os eventos em comum definidos são: ao final de (*onEnd*), ao início de (*onBegin*), ao pausar (*onPause*), ao retomar (*onResume*), e ao abortar (*onAbort*). As demais transições captadas são: ao ter sua apresentação travada (*onLock*) e destravada (*onUnlock*). Todas elas podem ser utilizadas na lógica das *Circunstâncias* e *Pontos de Decisão*.

A fim também de auxiliar na composição lógica, o modelo permite que seja verificado se um elemento encontra-se atualmente em um estado específico, como, por exemplo, se ocorrendo, pela expressão *isOccurring*. É possível também verificar a alteração de valores de propriedades por meio do evento *onChange*.

4.5 Resumo de Entidades, Hierarquia e Atributos de PST

A Tabela 5 lista as entidades, o relacionamento hierárquico entre elas e seus atributos, de forma a servir como uma referência resumida das definições do modelo PST.

Tabela 5 – Entidades, Hierarquia e Atributos do Modelo PST

Elementos	Atributos	Elementos Filhos
Itens Globais	id name class state properties	none
PervasiveStorytelling	import circumstances metadata	Episode
Episode	circumstances metadata	Experience
Experience	repeat refer circumstances isOptional	Sequence
Sequence	repeat delay syncedSeqGroup firstOnlySeqGroup anySingleSeqGroup refer isOptional preferenceDeviceType circumstances maxDur	Media, SensoryEffect, DecisionPoint
Circumstance	conditions	Nenhum
Media	repeat type src controls prefetch async refer devType dur delay	none
SensoryEffect	type refer	Nenhum
DecisionPoint	timeOut timeOutAction	Option
Option	condtion action	none

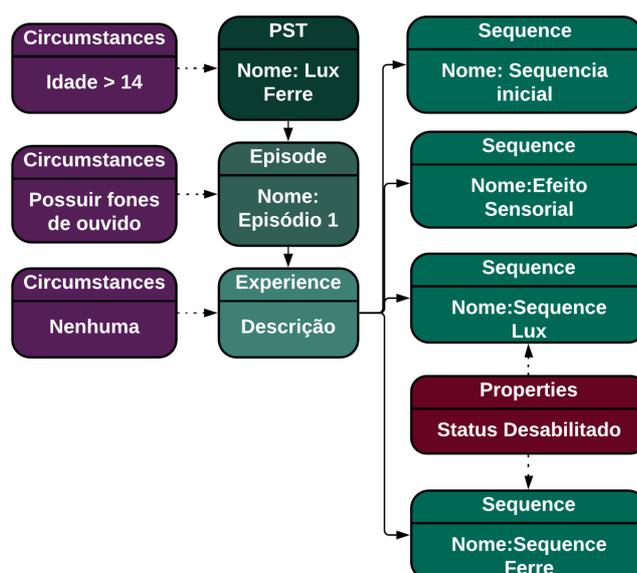
Fonte: Elaborada pelo autor (2021).

5 REPRESENTANDO UMA CONTAÇÃO DE HISTÓRIA COM O MODELO PST

Para avaliar a aceitação do modelo, pesquisadores do LMD/UFJF foram novamente reunidos, primeiramente para serem apresentados à versão final de PST, e, em seguida, para que elaborassem uma contação pervasiva de história seguindo as entidades definidas no modelo PST.

Devido a complicações de encontros presenciais geradas pela pandemia de Covid-19 em 2020/21, utilizou-se da ferramenta Lucidchart (31) para criação de um espaço de trabalho visual para equipes remotas. Através dela, foi possível trabalhar com os voluntários do grupo remotamente. A dinâmica consistiu na criação de uma narrativa através de blocos pré-montados de entidades definidas no modelo, sem auxílio de um pesquisador da computação na construção.

Figura 23 - Fragmento da Primeira Narrativa criada Pelo LMD - Pt. 1



Fonte: Elaborada pelo autor (2021).

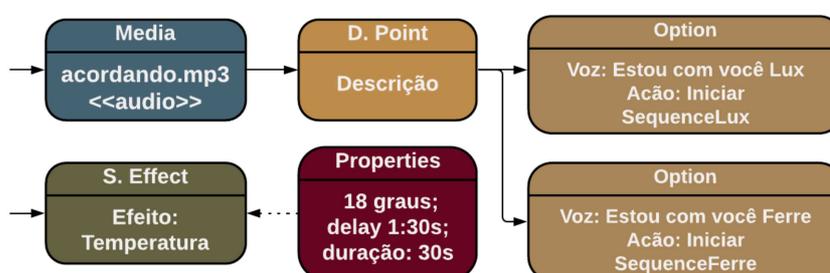
A descrição do conteúdo de cada elemento era de livre escrita para os participantes, dado que o objetivo era o que e como gostariam de representar um elemento em sua narrativa. Uma narrativa havia sido escrita previamente pelos participantes, no modelo convencional de roteiro redigido, mas já pensada para o uso de entradas por voz e controle de temperatura. A narrativa proposta conta a história de um padre que auxilia na investigação de um crime, tendo que lidar com variações do seu ego representados por entidades do *Bem* (Lux) e do *Mal* (Ferre), sendo estes os ganchos para tomadas de decisão por parte do interator.

Inicialmente, foi criado o primeiro elemento, definindo a instância de PST, nomeada com o nome da narrativa, “Lux Ferre”, e inserindo uma *Circumstance* de idade mínima para consumo, devido ao teor da narrativa. A ela foi associado, por

enquanto, um único episódio, o primeiro de uma série em construção para pesquisas de narrativas com uso de tecnologia de sons binaurais, existentes no grupo de pesquisa. Para o uso de tal tecnologia, um fone de ouvido é exigido na circunstância do episódio, para garantir que o fone esteja sempre presente.

Uma única experiência foi construída, e sem circunstâncias, por ser a única experiência a princípio desejada. Contudo, seria possível inserir outra experiência para caso o fone de ouvido presente tivesse suporte a áudio 7.1, reproduzindo com mídias específicas com efeitos de áudio 3D avançados, para uma melhor imersão e aproveitamento do equipamento disponibilizado. A representação visual elaborada do trecho citado neste parágrafo pode ser observado na Figura 23.

Figura 24 - Fragmento da Primeira Narrativa criada - Pt. 2

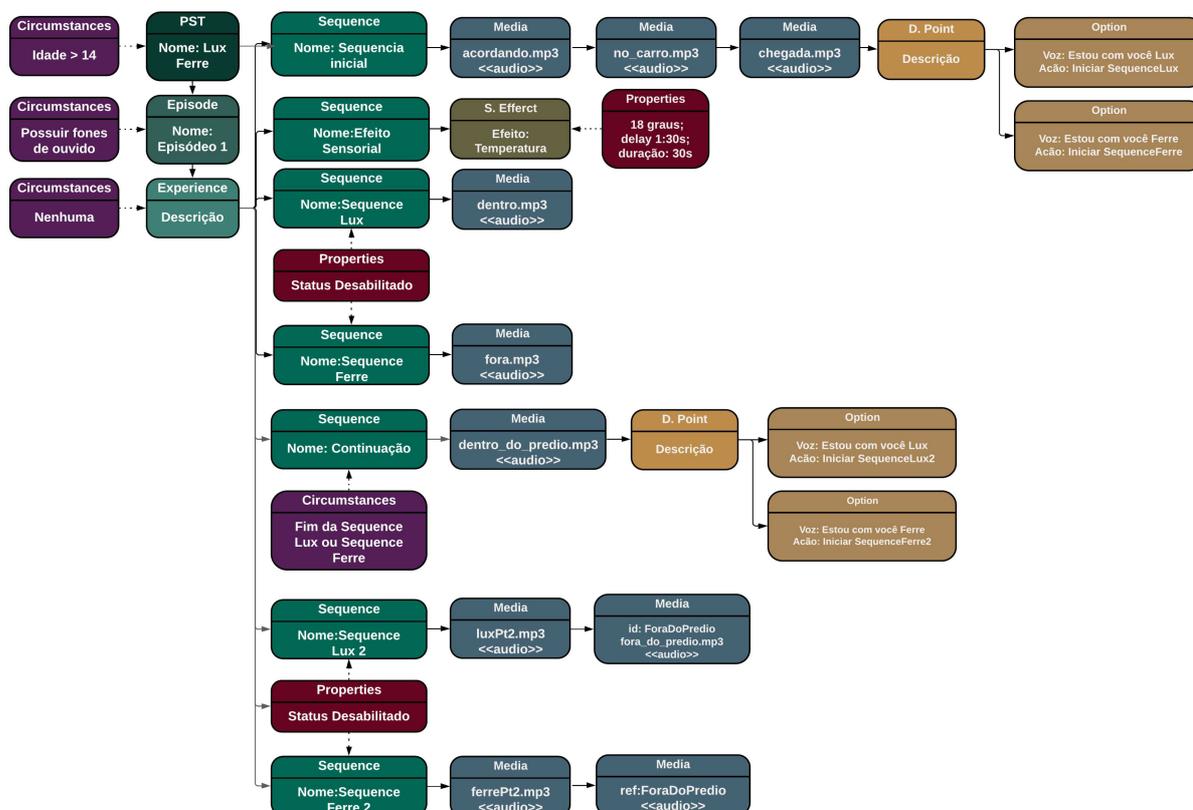


Fonte: Elaborada pelo autor (2021).

A experiência possui sete sequências, sendo uma inicial, cujo fragmento pode ser observado na Figura 24. Ela é composta por áudios produzidos para a narrativa e um ponto de decisão, em que o interator, após a reprodução dos áudios da sequência, deverá escolher um dos lados do alter-ego “Lux” ou “Ferre”. No ponto de decisão, entrada é captada por voz, cuja escolha da opção ativa a reprodução de outra sequência na experiência. Uma sequência com efeito sensorial de temperatura é executada em paralelo. No caso, os autores optaram no momento da criação por definir em graus a temperatura, e inserir um atraso que corresponde com o momento desejado de reprodução em referência ao áudio reproduzindo na sequência que executa em paralelo.

Uma das sequências é executada após a reprodução da sequência selecionada no primeiro ponto de decisão, sendo ela composta também por áudios e mais um ponto de decisão que dispara a reprodução de outra sequência. As demais sequências possuem atributo de estado desabilitado, pois são destravadas e executadas conforme as escolhas do interator nos pontos de decisão definidos pelo autor. A narrativa completa pode ser observada na Figura 25. Uma representação em JSON foi gerada para representar tal instância, com o fim de testá-la no protótipo do orquestrador. Tal descrição JSON pode ser encontrada no Apêndice A.

Figura 25 - primeira Narrativa criada Pelo LMD, Lux Ferre - Completa



Fonte: Elaborada pelo autor (2021).

5.1 PESQUISA EXPLORATÓRIA SOBRE *FEEDBACKS* E COMPREENSÃO DO MODELO PST

Uma pesquisa de natureza exploratória qualitativa, baseada conforme descrito por Toledo (59), foi desenvolvida e aplicada a pesquisadores voluntários do LMD, de modo a obter um parecer inicial referente a compreensão, ainda que superficial, dos profissionais da comunicação sobre as principais entidades do modelo e sua usabilidade. O estudo consistiu de questionários avaliativos, *feedbacks* verbais, e análise das informações coletadas por pesquisadores colaboradores da área de computação e comunicação.

Para explorar e analisar a compreensão do modelo ao longo da pesquisa e sua criação, ocorreram reuniões virtuais esporádicas distribuídas ao longo dos meses de novembro de 2020 e maio de 2021 com os integrantes do LMD, para apresentar a eles o estado vigente do modelo. Durante as reuniões, os participantes foram incentivados a manipular as entidades do modelo via Lucid Chart, para realizarem a construção de histórias próprias. Na última reunião, foram expostas às entidades finais do PST, junto a uma nova sessão de manipulação dos elementos para criação de uma narrativa. Ao fim, foi solicitado aos participantes do dia que respondessem um simples questionário (disponível no Apêndice B), contendo

duas perguntas. O questionário possui o objetivo de obter um *feedback* individual dos participantes sobre a compreensão a respeito das entidades. O questionário foi respondido por um total de nove integrantes. A primeira pergunta solicitava que o participante avaliasse em uma escala de 1 (*Não compreendi o modelo*) a 5 (*Compreendi por Completo*), sendo que cinco marcaram “[5] Compreendi por completo”, e os demais, no total de quatro, marcaram “[4] Compreendi a maior parte”. A segunda pergunta questionava se acreditavam que “o modelo ajuda na concepção de contações pervasivas de histórias frente ao que existe atualmente”, onde por unanimidade afirmaram positivamente.

Entretanto, foi perceptível em reuniões de sessões de teste de produção de narrativas com o modelo, uma grande dificuldade inicial na compreensão e aceitação do conceito da contação se basear em eventos, por partes dos mesmos membros que responderam o questionário. Os voluntários por vezes sentiam necessidade de representar uma linha do tempo, para poder representar as mídias e efeitos sensoriais pertencentes a uma sequência. Além disso, também desejavam que na interface visual representativa das entidades do modelo, que o último elemento de uma sequência pudesse apontar para outra sequência ou experiência, dando a “sensação de continuidade”, um após o outro, ao invés de fazerem o uso das circunstâncias com o pensamento de “eventualidade”. Quando utilizavam dos pontos de decisão para tomadas de decisão, se sentiram desconfortáveis com o fato de não haver um “elo visual direto” no Lucid Chart, entre a ação de cada opção e o elemento alvo da ação.

Efeitos sensoriais foram subutilizados. Grande parte das narrativas pensadas pelos voluntários eram focadas em mídias tradicionais, como o áudio. O desconhecimento de como utilizar as capacidades de efeitos gerados por múltiplos tipos de dispositivos também pôde ser evidenciada. Parte dos autores demonstravam-se a princípio “perdidos” em como utilizar para agregar a narrativa os efeitos sensoriais e pontos de decisão. Os pontos de decisão e circunstâncias foram utilizados somente para exigir que o autor interaja diretamente ou a presença de um dispositivo. Com isso, não foram elaborados alguns dos cenários imaginados na concepção do modelo, como uso de aura para proporcionar eventos únicos baseados no histórico do interator.

Após a análise supracitada, fora realizada uma nova reunião com parte dos membros do LMD que participaram da pesquisa, na qual os voluntários verbalmente expuseram suas dificuldades com relação à capacidade de compreender como produzir no cenário pervasivo, baseado em eventos. Relataram que o modelo é sim válido, mas exige dos autores uma nova forma de se pensar em relação ao que estão acostumados referente a suas experiências em criação de histórias, extrapolando a linearidade e sequencialidade. Dito isso, ressaltaram que, para um melhor uso da capacidade total do modelo, é necessário o treinamento de roteiristas de histórias lineares para o desenvolvimento do pensamento assíncrono e causal, além de melhor exemplificação do comportamento de efeitos sensoriais, e formas possíveis de se utilizar a aura e os pontos de decisão.

Com isso, ressalta-se a grande importância de um treinamento para criação de narrativas não lineares, e a necessidade de uma interface gráfica inteligente

capaz de auxiliar os autores na criação da narrativa. Além de intuitiva, a interface deve explicar aos autores como os elementos devem ser representados, sugerir efeitos sensoriais baseados nas mídias convencionais inseridas e sugerir mudanças de estrutura para melhor se adequar ao contexto de eventos, auxiliando os autores na familiarização do novo paradigma apresentado.

Devido à pandemia de Covid-19, infelizmente não ocorreu um número desejado de encontros e testes, para reforçar o modelo entre os voluntários, e criação de melhores exemplos com efeitos sensoriais, aura, pontos de decisão e circunstâncias. Acredita-se que mais encontros eram necessários, para que os autores tivessem capacidade de compreender, familiarizar-se com o ambiente proposto, e desenvolver a capacidade do uso dos elementos citados anteriormente, nas narrativas desenvolvidas.

Devido ao objetivo final definido no escopo deste trabalho em produzir um modelo de representação, que pode ser explorado em uma futura linguagem e ferramenta visual, este trabalho não possui um aprofundamento em pesquisas de avaliação de usabilidade do modelo. A presente subseção, conforme descrita, faz uma exposição do processo de uma pesquisa exploratória realizada para levantamento de *feedbacks* para apoio durante a produção do modelo. Uma avaliação aprofundada é deixada para trabalhos futuros, no contexto de uma ferramenta visual a ser desenvolvida, a qual demandará avaliação de uso por profissionais da área de comunicação.

6 Orquestrador Baseado em Nuvem

Uma máquina de apresentação, ou orquestrador, suportando as principais entidades do modelo PST foi prototipada, de modo a verificar a viabilidade de uma implementação capaz de lidar com instâncias do modelo e realizar a orquestração da contação pervasiva de uma história. Para que o protótipo pudesse compreender o que deve ser apresentado, optou-se por representar as instâncias do modelo através de uma estrutura em JSON predefinida e baseada na proposta do modelo constatada na Seção 4.2. É importante ressaltar que o presente trabalho não define linguagem, sendo a representação JSON realizada apenas com o objetivo de poder executar as representações visuais elaboradas pelos roteiristas voluntários em 5, as transpondo para um formato de fácil processamento para o orquestrador.

A estrutura JSON é bem simplificada. Cada instância de entidades estruturais é representada pelo seu respectivo nome ou sigla *pst*, *episode*, *experience*, *sequence*, *media*, *sensoryeffect* e *decisionpoint*, e os atributos são armazenados na chave “\$”. No objeto referente ao parâmetro de cada elemento ficam contidos as “circumstances”. Um exemplo de código JSON produzido e executado pode ser observado no Apêndice A.

Por se tratar de uma aplicação em nuvem, com requisições frequentes, ativações assíncronas e constante fluxo de requisições, a máquina foi desenvolvida para o ambiente de execução Node JS ¹ em Typescript ². Tais tecnologias propiciam diversos recursos nativos, além de bibliotecas consolidadas para soluções cliente-servidor assíncronas, e controladores de dispositivos IoT com APIs públicas em HTTP-REST (50) via serviço de nuvem de terceiros ou diretamente na rede local.

Uma arquitetura foi definida para o desenvolvimento da máquina de apresentação, para que cada elemento do modelo pudesse ser melhor representado e acoplado. A arquitetura possui similaridades ao proposto por (43), que desacopla as camadas de sensores, atuadores, comunicação em rede, processamento de dados e aplicação de controle e execução.

Conforme pode ser observado na Figura 26, há um controlador específico para lidar com requisições, controles e dados referentes a um interator *Interactor’s Aura Layer*. Para a orquestração da reprodução, um núcleo principal denominado *Presentation Engine* foi criado, seguindo as *Entidades Estruturais* e a *Máquina de estados das Entidades*. Para que múltiplos dispositivos pudessem ser controlados sem que a máquina de apresentação necessitasse se preocupar com variabilidade de dispositivos e afins, uma camada de abstração de controle de dispositivos sensores e atuadores foi implementada (*Device Control Abstraction Layer*).

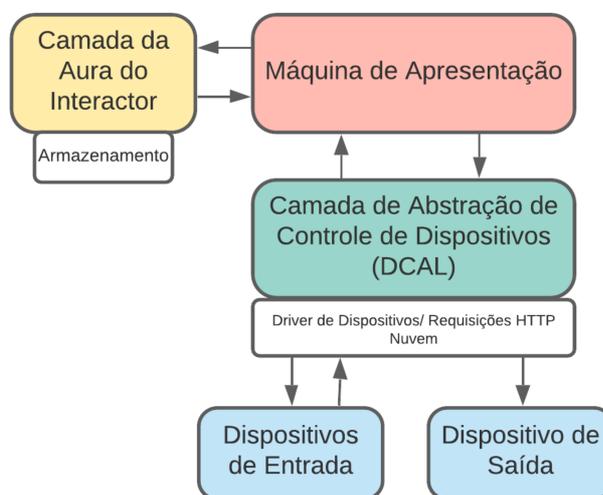
Todas as camadas aqui citadas respondem em HTTP-REST com corpo em JSON, sendo elas serviços desacoplados, mas capazes de manter uma comunicação. A API foi desenvolvida utilizando o framework Express ³, com uso de entidades controladoras e modelos, para cada funcionalidade implementada.

¹ <https://nodejs.org/en/>

² <https://www.typescriptlang.org/>

³ <https://expressjs.com/>

Figura 26 - Arquitetura da Máquina de Apresentação Baseada em Nuvem



Fonte: Elaborada pelo autor (2021).

6.1 INTERACTOR'S AURA LAYER

Para poder representar, controlar e manipular os dados de um interactor e consultas de *Aura* solicitadas pela máquina de apresentação, foi implementada uma camada dedicada à entidade *Interactor*. Através deste módulo, a máquina de apresentação consegue persistir e consultar as instâncias de contação de histórias (*PST Instances*) às quais o interactor se encontra inscrito. Toda a estrutura segue conforme definido na Seção 4.3.

Quando um interactor deseja participar de uma nova história, uma requisição deve ser feita à máquina de apresentação, informando qual *PST* e qual *interactor* estão envolvidos. Assim, uma nova instância da *PST* é criada e associada ao interactor em sua aura. Uma vez iniciada, todas as alterações referentes ao percurso narrativo vivido (*Aura*) daquele interactor serão armazenadas nesta instância. Nessa camada também se encontram e podem ser consultadas, quando necessário pela narrativa, as informações do interactor, como preferências e dispositivos registrados (*Properties*).

6.2 DEVICE CONTROL ABSTRACTION LAYER

A prototipação incluiu o suporte a cinco dispositivos inteligentes conectados em rede como atuadores para geração de efeitos, sendo estes, duas lâmpadas inteligentes, uma da marca Xiaomi Yeelight e outra Tuya, além de um controle infravermelho e tomadas inteligentes, ambos também da Tuya. Para sensoriamento e captação de *inputs*, utilizou-se de um dispositivo móvel, através de uma aplicação multiplataforma (Android ou iOS) produzida com o *framework* React-Native, capaz de pegar geolocalização em tempo real e captura por voz em *speech-to-text*.

Figura 27 - Dispositivos de saída utilizados na pesquisa.



Fonte: Elaborada pelo autor (2021).

Nota: Primeira lâmpada a esquerda da Yeelight Xiaomi.
Demais dispositivos da Tuya, sendo em ordem uma lâmpada, controlador infravermelho e duas tomadas

Efeitos que não podiam ser reproduzidos com dispositivos disponíveis, ou objetos de mídia reais não produzidas, tiveram sua apresentação representada textualmente por logs no terminal do serviço de nuvem em que o orquestrador foi instalado e colocado em execução.

Cada dispositivo e serviço de nuvem IoT apresentou diferentes formas e APIs para comunicação. A variabilidade para comandos similares como “ligar/desligar” entre dispositivos de mesma marca, também foi encontrada. Para facilitar a comunicação entre os dispositivos, uma camada de abstração foi criada, a *Device Control Abstraction Layer* (DCAL). Através dela, é possível requisitar, por comandos padronizados, que uma categoria de efeito seja reproduzido, fazendo uso de um dispositivo padrão, ou podendo definir através de IDs, ou apelidos dados pelo próprio interator registrados em sua *Aura*.

A camada de abstração consegue interpretar os padrões de comandos solicitados pela máquina de apresentação, e distribuir aos dispositivos desejados, através de *drivers* implementados na camada. Com isso, remove-se a necessidade de as demais camadas saberem se comunicar com os dispositivos em uma aplicação. Processo esse similar ao conceito do *middleware* proposto por (8) para controle de dispositivos IoT de cuidado pessoal, com a diferença de que não há necessidade de que este execute na mesma máquina em que se encontra a máquina de apresentação. Com isso, assistentes virtuais, ou controladores de grupos de dispositivos, ao receberem a solicitação, conseguem distribuir as ações entre seus dispositivos disponíveis.

A camada foi implementada seguindo o padrão REST com solicitações HTTP e corpo da mensagem em JSON, um padrão similar ao que foi encontrado nas APIs de controle de dispositivos. Na URI da requisição deve ser informado o nome (apelido) dado pelo interator, ou id real informado pelo próprio dispositivo em que deseja obter informações, ou requisitar controle. Esses dados devem ser fornecidos pela

Figura 28 - Console da Máquina de Apresentação.

```

13:46:39 - Found 0 errors. Watching for file changes.
[[13:46:40.202]] [LOG] server started at http://localhost:10100
[[13:46:41.816]] [LOG] Starting.. 3c802c10-a9e4-4a0d-b9c8-b2cff35d82e7
[[13:46:41.817]] [LOG] Iniciando na engine: pst1 Meu Pai:none
[[13:46:42.823]] [LOG] Im: pst1 - Status Changed -> occurring
[[13:46:42.823]] [LOG] Iniciando na engine: primeiroEpisodio Meu Pai:pst1
[[13:46:43.830]] [LOG] Im: primeiroEpisodio - Status Changed -> occurring
[[13:46:43.830]] [LOG] Iniciando na engine: exp1 Meu Pai:primeiroEpisodio
[[13:46:44.845]] [LOG] Im: exp1 - Status Changed -> occurring
[[13:46:44.845]] [LOG] Iniciando na engine: sequenciaInicialexp1 Meu Pai:exp1
[[13:46:44.845]] [LOG] Iniciando na engine: sequenciaSensorialTemperatura Meu Pai:exp1
[[13:46:44.846]] [LOG] Iniciando na engine: SegundaSequencia Meu Pai:exp1
[[13:46:45.853]] [LOG] Im: sequenciaInicialexp1 - Status Changed -> occurring
[[13:46:45.854]] [LOG] Iniciando audioacordando
[[13:46:45.856]] [LOG] Im: sequenciaSensorialTemperatura - Status Changed -> occurring
[[13:46:45.856]] [LOG] Iniciando sensorialEffectlexp1
[[13:46:50.869]] [LOG] Encerrando audioacordando - Next: audiono_carro
[[13:46:50.869]] [LOG] Iniciando na engine: audiono_carro Meu Pai:sequenciaInicialexp1
[[13:46:50.915]] [LOG] Im: sequenciaSensorialTemperatura - Status Changed -> finished
[[13:46:51.873]] [LOG] Im: audiono_carro - Status Changed -> occurring
[[13:46:51.873]] [LOG] Iniciando audiono_carro
[[13:46:56.881]] [LOG] Encerrando audiono_carro - Next: audiochegada
[[13:46:56.881]] [LOG] Iniciando na engine: audiochegada Meu Pai:sequenciaInicialexp1
[[13:46:56.929]] [LOG] Im: audiochegada - Status Changed -> occurring
[[13:46:56.929]] [LOG] Iniciando audiochegada
[[13:47:01.930]] [LOG] Encerrando audiochegada - Next: dp1exp1
[[13:47:01.931]] [LOG] Iniciando na engine: dp1exp1 Meu Pai:sequenciaInicialexp1
[[13:47:01.963]] [LOG] Im: dp1exp1 - Status Changed -> occurring
[[13:47:01.963]] [LOG] Iniciando dp1exp1

```

Fonte: Elaborada pelo autor (2021).

Nota: Console exibindo a orquestração de uma instância de PST sem utilizar mídias prontas ou dispositivos conectados, apenas as representando.

camada do iterador.

O método GET retorna dados de um dispositivo, como seu estado atual e respectivas configurações em atividade. Em caso de ser um dispositivo de entrada, obtém e retorna valores de *inputs* do sensor especificado. O POST é utilizado para a criação de registro de dispositivos. Deve ser informado em seu corpo um JSON com “id”, nome (apelido), tipo do dispositivo e a marca para que o protótipo saiba com qual *driver* lidar. O método PUT é utilizado para solicitar ações sobre um dispositivo, onde as novas configurações devem ser passadas através de um corpo em JSON com as instruções.

No momento em que chega uma solicitação à DCAL, ela é processada e identificada qual ação foi solicitada. Em caso de um *output*, o *driver* interpreta as configurações solicitadas e traduz para o formato do dispositivo, o qual efetuará a saída, e realiza a solicitação, retornando ao orquestrador se foi ou não possível executar. Em caso de um *input*, também é processada a forma de comunicação com o dispositivo, contudo, nesse caso é processada a resposta para um formato padronizado predefinido antes de responder ao orquestrador.

O aplicativo para captura de dados foi desenvolvido para *smartphones* Android e iOS, com uso da tecnologia React-Native, que permite exportar uma mesma aplicação para ambos os sistemas. Por questões de privacidade e segurança, aplica-

ções para esses sistemas necessitam de autorização de uso de certas APIs por parte do usuário. Em sua primeira execução, a aplicação solicita ao usuário que forneça permissão para compartilhamento de geolocalização e captura de voz, sendo estes os *inputs* utilizados na primeira versão do aplicativo.

Figura 29 — Tela do aplicativo Mobile de captura de entrada.



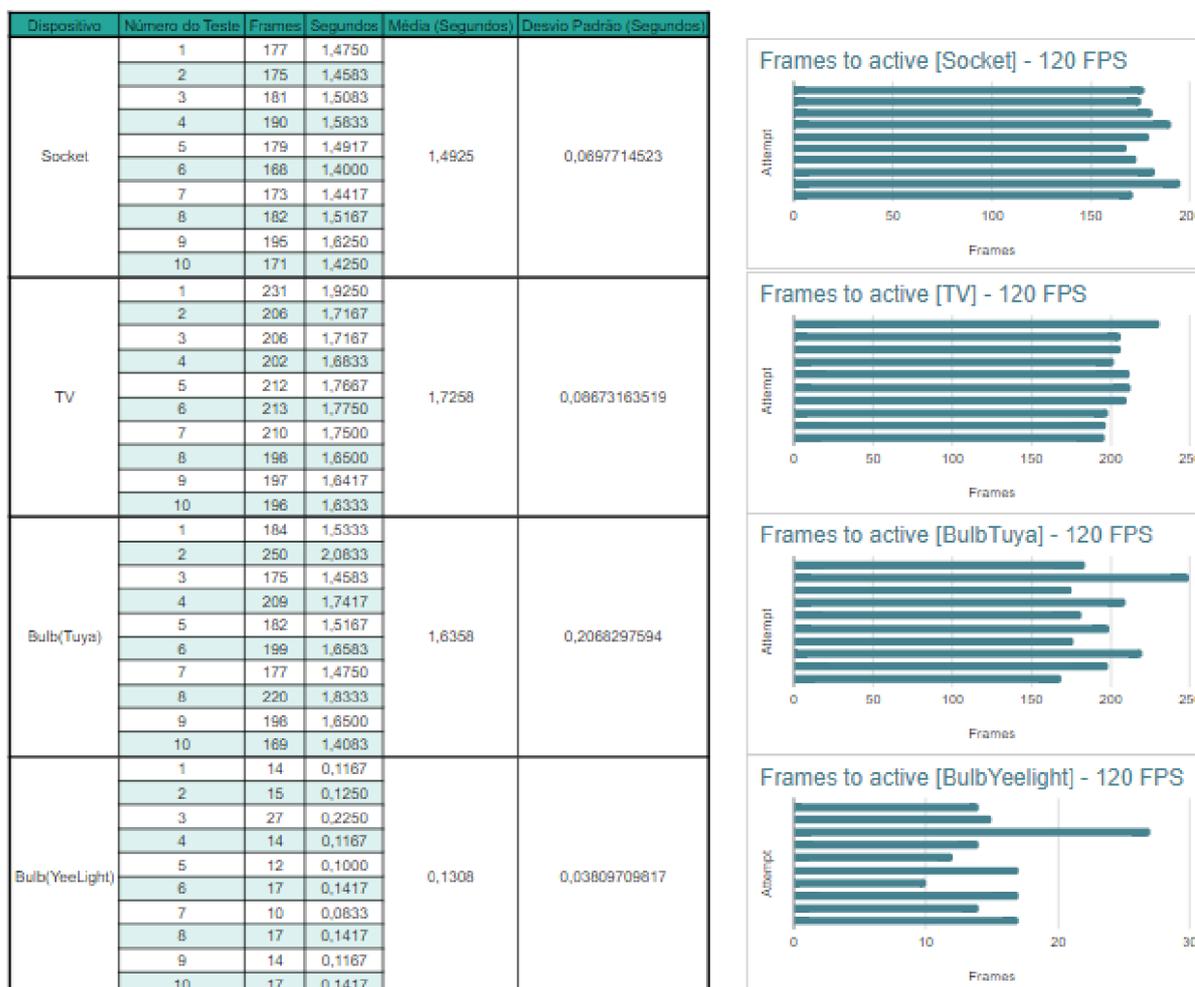
Fonte: Elaborada pelo autor (2021).

Conforme pode ser observado na Figura 29, a aplicação é visualmente simples, com botões apenas para caso não sejam fornecidas inicialmente as permissões, o usuário possa permiti-las mais tarde. Quando se obtém as informações referentes à geolocalização, é exibida na tela uma mensagem alertando sobre a obtenção. Para a captura de voz, além da mensagem na tela solicitando que o usuário fale, também é tocado um alerta avisando o início da gravação, e um som para alertar o fim.

A DCAL também foi testada por outras aplicações além do orquestrador aqui proposto e implementado. Através de um *chatbot* capaz de capturar textos de um *chat* em um canal específico da plataforma Twitch.tv, que transmitia em tempo real um ambiente com os dispositivos, espectadores podiam mandar comandos por texto sinalizados pelo carácter inicial "!", nome do dispositivo e ação, para os controlar. As informações eram processadas pelo *chatbot*, e a requisição via HTTP era feita ao DCAL.

Durante esse teste, a camada era colocada sob constante requisição, e os dispositivos lidando com várias novas solicitações. Em sua versão final, o DCAL conseguiu orquestrar sem falhas, mesmo em cenários com múltiplas requisições em pouco tempo. A transmissão era realizada em baixa latência, onde perdas eram aceitáveis para garantir a entrega no menor tempo possível, o que poderia

Figura 31 — Resultados do teste de atraso de execução dos dispositivos - Local.



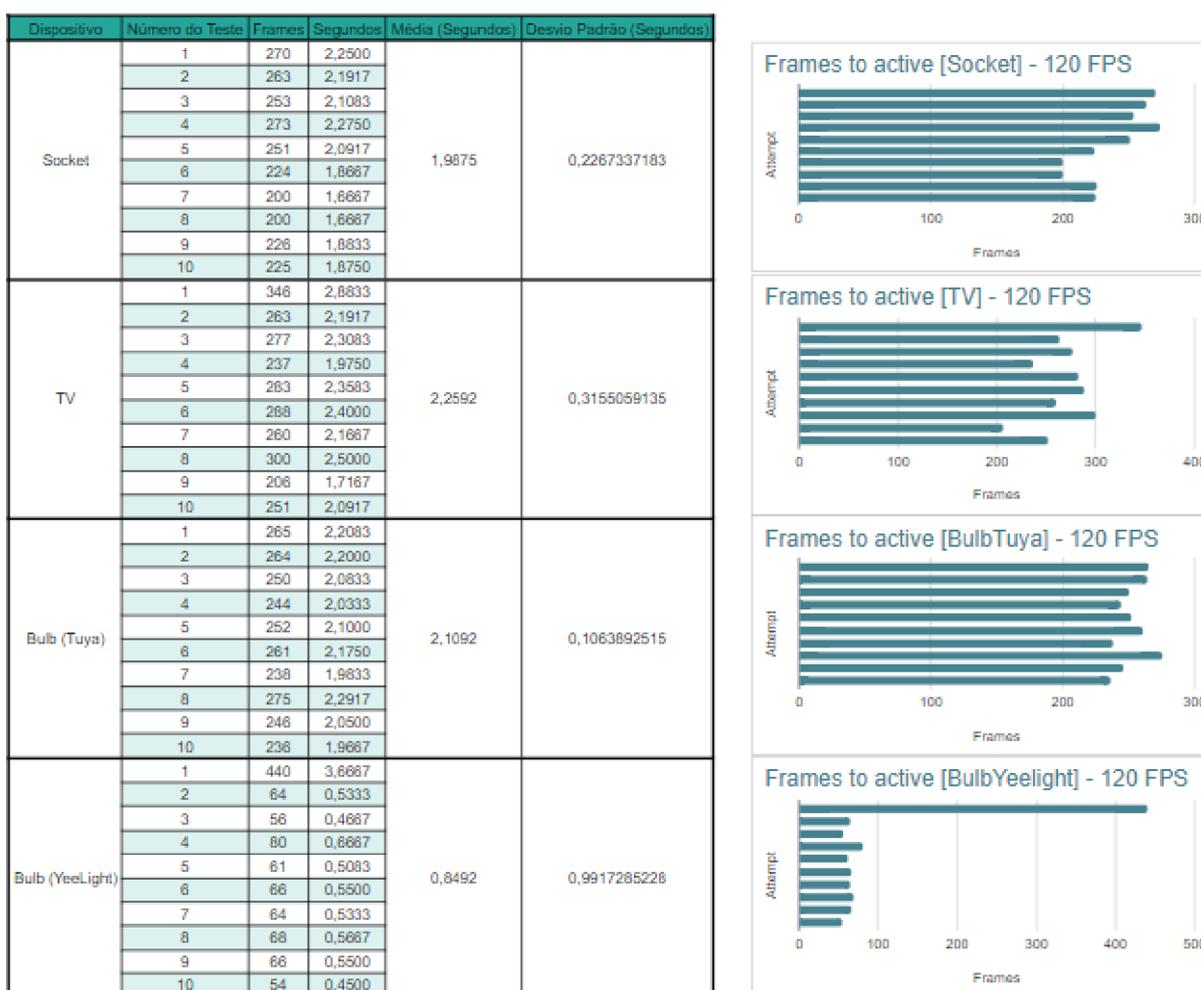
Fonte: Elaborada pelo autor (2021).

no datacenter RePesq da UFJF, a nove saltos da máquina cliente.

O teste consistiu na criação de uma instância de PST com uma sequência unicamente formada por dez (10) elementos de efeitos sensoriais com uma ação destinada a um dispositivo, que eram executados a cada 5 segundos. Para calcular o *delay*, uma câmera capturou a 120 quadros por segundo o momento em que um *log* comunicando o envio da solicitação ao dispositivo era exibido no console do protótipo, e o momento em que o dispositivo gera o efeito final. Com uso de um *software* profissional de edição de vídeos, obteve-se cada quadro entre os acontecimentos, calculando-se o valor total do tempo gasto. O resultado dos testes, assim como a média dos atrasos pode ser observado na Figura 31.

A Figura 31 contém os dados do teste executando com o protótipo do orquestrador na mesma rede em que se encontram os dispositivos de teste. Já a figura 32, possui os dados referentes à execução do orquestrador em nuvem. Em um comparativo entre os dispositivos da marca Tuya (Lâmpada, TV e Tomada),

Figura 32 — Resultados do teste de atraso de execução dos dispositivos - Nuvem.



Fonte: Elaborada pelo autor (2021).

em que eram controlados pela nuvem da fabricante, a máquina executando em nuvem apresentou um delay um pouco maior, mas em ambas a média foi superior 1,49 segundos, o que pode ocasionar certa frustração na qualidade da experiência do usuário dependendo do tipo de efeito, necessitando de um melhor controle de sincronia e preparação. Em relação ao dispositivo da YeeLight com um controlador disponível na borda, mesmo sem remover os *outliers*, todos os testes apresentaram um tempo inferior a um segundo, tanto com o controlador em rede local ou em nuvem, sendo este um possível indício de que pode-se priorizar o uso de dispositivos em borda (*edge computing*) (53) para o cenário de contação pervasiva de histórias com uso de dispositivos IoT.

6.3 MÁQUINA DA APRESENTAÇÃO

A máquina de apresentação, *Presentation Engine*, controla e solicita a execução de todos os itens de uma instância de PST a ser consumida, que segue a

estrutura definida na Seção 4.2, com sua lógica de reprodução conforme a Seção 4.4.

Além de controlar a apresentação, a Máquina de Apresentação pode ser controlada externamente, através de requisições em HTTP-REST. Para execução das ações definidas no modelo que podem ser observadas na Figura 22, como, por exemplo, *start*, *pause*, *abort*, foi criada uma rota na API para a entidade denominada *Presentation Engine*, capaz receber as requisições e processar a solicitação na nuvem, simulando assim ferramentas de controle de reprodução que um interator teria. Através dela, passando os parâmetros definidos na rota e corpo da solicitação, é possível iniciar o episódio de uma *PST* já associada a um interator, registrar novas *PSTs*, alterar dados do interator, entre outras funcionalidades.

Com fins de teste, a API desenvolvida para conversar com a máquina de apresentação permite também alterar valores do interator ou da *PST* em tempo de execução, assim como forçar a mudança de estados, com a finalidade de verificar o comportamento da máquina diante de situações possíveis de ocorrer durante uma apresentação, como ações de um interator sobre a apresentação (*start*, *stop*, *pause*, *resume*), ou possíveis modificações que viriam de outros *runtimes*. A máquina de apresentação produzida se mostrou incapaz de quebrar regras de transição mesmo quando forçada, como, por exemplo, abortar um elemento em estado *Enabled*, seguindo corretamente a definição da Máquina de Estados das Entidades 4.4.

No momento em que a máquina de apresentação é chamada por um interator para iniciar um episódio, a instância da instância de *PST* a qual pertence, é resgatada da camada da aura do interator, e passada para uma função de inicialização que verifica suas circunstâncias (conforme 4.2.5), para validar se a história pode ocorrer. Em caso positivo, a *PST* e o *Episode* selecionado são iniciados (*start*) e registrados em uma nova *thread* da *engine* capaz de observar futuras alterações. Essa *engine* é responsável por alertar quando um elemento deve ter suas circunstâncias novamente avaliadas, chamando uma função de avaliação para o elemento registrado.

Para este trabalho, duas versões da *engine* foram implementadas, uma baseada em tempo, e outra por observação de alteração. A baseada em tempo, funciona com um temporizador predefinido, em que a cada X milissegundos definidos pelo programador, a função de avaliação é chamada. Neste modelo, reavaliações eram feitas periodicamente e, na maior parte das vezes, de forma desnecessária.

Na segunda versão, é criado um observador de alterações na base de dados da *Aura*, na máquina de estados da apresentação e nos dispositivos registrados. Deste modo, cada *thread* criada observa todos os elementos citados, chamando a função de avaliação caso alguma alteração ocorra. Apesar de reduzir verificações desnecessárias em relação à verificação periódica, *threads* da *engine* que não são afetadas pelas mudanças de um elemento específico, ainda assim são executadas, e tem suas circunstâncias reavaliadas.

Um terceiro modo foi elaborado, mas não teve sua implementação finalizada para testes. Neste formato, cada *thread* deve observar unicamente e individualmente itens de aura, estados e dispositivos que a ela condiz constantemente, aguardando que sofra alguma alteração, criando um observador para cada elemento existente na lógica de uma circunstância.

Independente da forma e momento em que uma circunstância é avaliada, toda circunstância passa por um mesmo processo de avaliação. A função de avaliação verifica inicialmente o estado do pai, devido à indeterminação de quando o mesmo pode parar de ocorrer, e precisa estar ocorrendo para que os elementos filhos também ocorram. Como exemplo, um elemento em estado *Enabled*, quando habilitado por um ponto de decisão, verifica se o pai está ocorrendo, e posteriormente verifica as circunstâncias para ocorrer.

Quando um elemento está ocorrendo, *Occurring*, a *engine* precisa verificar constantemente o estado do pai e as circunstâncias para continuar ocorrendo. A *engine* de avaliação também verifica os estados dos filhos, para observar se os mesmos já foram encerrados, e assim garantir seu fim natural. Caso um elemento seja desabilitado ou encontre o fim natural, o mesmo é removido da *engine*, e precisa ser instanciado novamente, caso novamente habilitado.

O processo de habilitação permite que um ponto de decisão solicite que uma entidade seja habilitada pela primeira vez ou novamente, estando ela desabilitada (*Disabled*) ou finalizada (*Finished*), conforme mencionado em 4.4.

Tendo apresentado a estrutura de funcionamento da *engine*, em suma, um episódio ocorrendo deve ter verificado se os seus filhos *Experiences* estão habilitados a terem circunstâncias avaliadas, e, em caso positivo, tê-los registrados na *engine*. A partir disso, cada experiência, após sofrer a transição de *start*, se bem sucedida, passa pelo mesmo processo anterior, registrando cada um de seus filhos *Sequences* em uma nova *thread* na *engine*, caso habilitadas.

No caso de entidades de entrada e saída, como objetos de mídia, efeitos sensoriais e pontos de decisão, a máquina de apresentação não as instancia em uma nova *thread* da *engine* como os demais elementos. Uma função de execução é chamada, aciona a apresentação do elemento, e observa constantemente o estado do elemento-pai, em caso de alguma alteração, o objeto sofre a ação cascadeada.

Quais objetos serão utilizados na apresentação e respectivas propriedades são decididos em conformidade com a camada do *Interactor*, conforme o efeito sensorial e mídia descrita na instância de PST. Os comandos para objetos de entrada e saída, são então enviados via HTTP-REST para a camada de *Device Control Abstraction Layer*, para que a mesma possa processar os efeitos sensoriais e entradas, e retornar quando a apresentação foi possível e/ou resultados das entradas necessárias.

6.4 EXEMPLOS DE EXECUÇÃO

Para poder validar o funcionamento da máquina, uma série de PSTs compostas somente com os elementos necessários para testar a arquitetura e os dispositivos foi produzida e submetida para a máquina orquestrar. Por possuir os principais elementos do modelo, a primeira apresentação utilizada para teste foi a “Lux Ferre”. Com ela foi possível testar a capacidade do orquestrador em nuvem de registrar um novo *Interactor* e uma nova instância de PST no banco de PSTs. Posteriormente, o mesmo também conseguiu associar a PST armazenada ao novo *Interactor*.

A PST “Lux Ferre” é composta por um *episode*, uma única *Experience* e sete *Sequences*, sendo essas com *Circumstances* variadas e *Decision Points* capazes de

liberar sequências a serem reproduzidas.

Quando é enviada uma solicitação de “start” para a *Máquina de Apresentação*, é informado junto o *Interactor*, o ID da PST registrada e qual *Episode* será iniciado. A *Máquina de Apresentação* então verifica na *Camada da Aura do Interactor* se existe tal registro de PST associada ao interator definido. Sendo verdadeiro, carrega e verifica se está apta para iniciar. No caso, é definido que o interator possua ao menos 14 anos. Essa informação é então solicitada à *Camada da Aura do Interactor*, e se verifica se a idade respondida condiz com a mínima definida. Sendo verdade, então são verificadas as circunstâncias do episódio selecionado, que, no caso, exige o uso de um fone de ouvido. Como os áudios não foram produzidos de fato, ao consultar a *Camada de Abstração do Controle de Dispositivos*, a mesma foi programada para responder que existe a presença do dispositivo associado ao interator informado.

Sendo então ambas circunstâncias verdadeiras da PST e do Episódio selecionado, são registrados na *engine* de verificação de circunstâncias. Com isso, a máquina inicia a verificação da experiência registrada. É dado então início a *Experiência*, e suas sequências são verificadas. De início, há somente duas sequências à habilitadas, uma composta por três arquivos de áudio e ponto de decisão, e outra por um efeito de temperatura. As demais encontram-se desabilitadas para não ocorrer de início.

Como saída das mídias de uma das sequências habilitadas foram especificados arquivos de áudio que infelizmente não puderam ser produzidos, por não se ter desenvolvido *drivers* para dispositivos de áudio. De qualquer forma, os itens foram representados textualmente no terminal do orquestrador no momento correspondente a sua reprodução, e com duração predefinida. Para entrada de dados do ponto de decisão, era solicitada a captação de voz de um interator. Para tal, utilizou-se de o aplicativo mencionado anteriormente, em um *smartfone* Android, capaz de receber solicitações, capturar a voz, e devolver ao orquestrador. Após obter o resultado da voz, é processada qual opção tem sua lógica validada, e sua ação ocorre, habilitando uma entre duas sequências que se encontram desabilitadas, “SequenceFerre” ou “SequenceLux”. O Controlador Infravermelho permite criar sub-dispositivos, como controle de TV, controle de ar-condicionado, entre outros. Com isso, foi criado um sub-dispositivo de controle de ar-condicionado para poder receber comandos referentes a efeitos sensoriais de temperatura.

A sequência habilitada então é iniciada e reproduzida, sendo ambas compostas por um único áudio. Após sua reprodução, outra sequência, cujas circunstâncias explicitam o fim de “SequenceFerre” ou “SequenceLux”, é iniciada, com um áudio e um ponto de decisão de composição similar ao anterior, também captando voz e liberando uma entre duas sequências finais. Após a reprodução da sequência final habilitada, “SequenceFerre2” ou “SequenceLux2”, a experiência compreende que encontrou seu fim natural, por todas as sequências estarem ou desabilitadas ou encerradas. O efeito em cascata passa para o *episódio*, e por consequência à PST, encerrando a apresentação. Vale ressaltar que todas as modificações de estado e ações são salvas na aura do interator no momento em que ocorrem.

7 CONCLUSÃO

O presente trabalho apresentou o modelo PST, proposto para criações de contações pervasivas de histórias, com ênfase em autores profissionais na criação de conteúdos. O modelo visa elevar a capacidade de abstração já encontrada em modelos como NCM/NCL e STorM/STorML, e uni-la a novos paradigmas de representação e obtenção de informação entre mundo real e o virtual, como os presentes no MPEG-V. Além disso, o mesmo aproveita dos benefícios das composições com semânticas incorporadas, como as utilizadas em SMIL, com a expressividade do paradigma de causalidade utilizado em NCL.

Infelizmente, as condições pandêmicas impossibilitaram avanços maiores sobre a avaliação de usabilidade do modelo pelo público ao qual se destina, necessitando de pesquisas futuras. No entanto, por tal fato, este trabalho pôde investir na investigação da capacidade do modelo de servir de base para gerar um ferramental apto à orquestração e à apresentação de uma história contada de forma pervasiva. A máquina de apresentação prototipada e executada em nuvem conseguiu orquestrar, apresentar, lidar com múltiplos dispositivos, e representar objetos de mídia e efeitos que compunham a narrativa desenvolvida pelo Laboratório de Mídia Digital, assim como testes forçados de reprodução.

Quanto à reprodução dos conteúdos, percebe-se a importância da agregação de recursos de computação de borda, principalmente no que diz respeito à qualidade de experiência do usuário final. É perceptível, em um primeiro momento, a discrepância no tempo de resposta entre requisição e efeito comparando um dispositivo em rede local, a um em serviços de nuvem. Tal resultado ressalta a necessidade de estudo de sincronismo da apresentação com uso de dispositivos comandados remotamente em rede. Tais conclusões deste e de trabalhos futuros vão ajudar na criação de boas práticas para a criação de contações pervasivas, que identifiquem de forma realista as expectativas de granularidade de sincronismos e tempos de respostas.

O presente trabalho omite parte das propriedades de elementos de mídias e efeitos sensoriais, devido ao escopo não se tratar de uma linguagem, e sim de um modelo conceitual, deixando apenas sugestões de implementação de tais propriedades, baseando-se em MPEG-V, NCL e STorML. Contudo, o modelo já deixa base para criação de estilos de apresentação por arquivos referenciados, de forma análoga ao HTML, ao utilizar CSS3 (63), e permitir criação de classes, nomenclaturas e identificadores. No desenvolvimento de uma futura linguagem, deseja-se, então, não somente a incorporação de estilos para mídias convencionais, mas também uma adaptação das representações de Efeitos Sensoriais observados no MPEG-V para mapeamentos de mais alto nível.

O modelo PST incorpora muitas contribuições de outros modelos hipermídia, linguagens e editores gráficos apresentados neste trabalho. Algumas das incorporações e evoluções:

- Incorporação de composições com semântica presentes nas diversas linguagens tais como semântica de:

- Sequenciamento previsto no <seq> de SMIL e trilhas do STorM, incorporado em *Sequences*, expandido para ser dependente de circunstâncias;
 - Paralelismo previsto no <par> de SMIL e Cenas do STorM, permitindo escopo definido ou não, incorporado em *Experiences* e *Episodes*, expandido para ser dependente de circunstâncias;
 - Reprodução excludente previsto no <excl> de SMIL, adaptado sob a forma de circunstâncias;
 - Relacionamentos causais personalizados, existentes em links/conectores de NCL e Trilhas interativas do STorML, simplificados sob a forma de pontos de decisão e circunstâncias;
- Máquina de Estados da Apresentação composta pelos principais estados e eventos de transição de NCL e STorML, expandidos para melhor se adaptar às variabilidades de ambientes pervasivos;
 - Mídias e respectivos atributos e propriedades baseados nas definições de NCM e STorM e HTML5;
 - Representação de Efeitos Sensoriais e Inputs baseados no MPEG-V;
 - Recurso de interações multimodais, presentes em evoluções de NCL e STorML.

Desvantagens do modelo em relação aos trabalhos supracitados:

- Pouco reúso quando comparado à NCL;
- Maior complexidade de especificação de sincronismo, por meio de circunstâncias;
- Aninhamento limitado de composições, se comparado ao NCM;
- Generalista, por ser modelo conceitual, em comparação a linguagens de domínio específico, otimizadas para um subgrupo de mídias e dispositivos de apresentação;
- Modelo com alto grau de abstração, necessitando de uma máquina de estados de apresentação para possibilitar o seu uso.

7.1 Contribuições da Dissertação.

Resumidamente, as principais contribuições do presente trabalho são:

- Criação de um modelo para representação de contações pervasivas de histórias, com uma estrutura representativa de elementos (entidades) intuitiva ao público alvo. Em pesquisa preliminar, autores que experimentaram a criação com o modelo reagiram positivamente;

- Modelo de acoplamento fraco entre estrutura e dispositivos, de modo a permitir uma flexibilidade de especificação, mantendo uma mesma estrutura para definir entrada e saída, apenas se adequando ao efeito de interação pretendido a cada elemento. Incorpora a capacidade de representação de Efeitos Sensoriais simplificada;
- Representação de conteúdos interativos abordando suas múltiplas modalidades de interação;
- Orquestrador de contações de histórias baseado em nuvem, onipresente e que não estabelece um dispositivo principal de apresentação;
- Adequação de uma máquina de estados de reprodução de entidades estruturais, para lidar com as variabilidades de ambientes pervasivos;
- Modelo adequado para a implementação de serviços de orquestração seguindo os paradigmas recentes de computação em nuvem, de borda e IoT, que permeiam atualmente a internet.

7.2 Trabalhos Futuros

O presente trabalho serve como base para o desenvolvimento de uma ferramenta de autoria visual, que possibilite aos profissionais do domínio especificado, desenvolverem conteúdos que permitam a exportação automática à máquina de apresentação em nuvem, prontos para consumo, evitando assim a necessidade de conhecimentos específicos de programação.

Além disso, o presente trabalho abre possibilidades de aprofundamento e maiores explorações em trabalhos futuros, tais como:

1. Desenvolvimento de uma linguagem textual para o modelo e execução na máquina de apresentação;
2. Aperfeiçoamento de uma sintaxe de intercâmbio baseada em JSON, para especificação da instância do modelo, para facilitar a integração com tecnologias de reprodução e ferramentas gráficas;
3. Desenvolvimento de uma ferramenta gráfica para automatizar a criação de conteúdos PST, oferecendo recomendações de inteligência artificial para a criação de experiências e interlocução com outras histórias;
4. Criação de um assistente inteligente incorporado à ferramenta gráfica capaz de verificar possíveis problemas na narrativa (como pausas irrecuperáveis), sugerir possibilidades ainda não exploradas, e itens que melhor se adéquem na história em produção;
5. Realizar mais pesquisas sobre sincronismo, latência e distribuição de dispositivos na rede coordenados por serviços de orquestração que se estendem da nuvem para a borda da rede;

6. Realizar pesquisas com autores em busca de indicadores de legibilidade, usabilidade, eficácia do modelo e futura ferramenta gráfica;
7. Evolução da máquina de apresentação para suporte à capacidade de multiusuário nos mesmos dispositivos, diferenciando interatores e armazenando nas respectivas auras;
8. Implementação de um orquestrador distribuído público para que pessoas possam registrar suas narrativas e as executar.

REFERÊNCIAS

- 1 ABNT. Norma NBR 15606. “**Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital**”, Rio de Janeiro, Brasil, 2018 - Disponível em: <https://www.abntcatalogo.com.br/norma.aspx?ID=406767>. Acessado em: 29 de jan. 2021
- 2 ALLEN, James F. **Maintaining knowledge about temporal intervals**. Communications of the ACM, v. 26, n. 11, p. 832-843, 1983.
- 3 AMAZON. 2021. **Alexa Skills Kit - O que são Alexa Skills?** Disponível em: <https://developer.amazon.com/pt-BR/alexa/alexa-skills-kit>. Acesso em: 7 de jun. 2021
- 4 AMAZON. 2021. **Create Alexa skills in minutes**. Disponível em: <https://blueprints.amazon.com>. Acessado em: 6 de jun. 2021
- 5 ARAMBURU, Rodrigo. (2012) **HTML5 novos elementos de estrutura**, Disponível em: <https://www.botecodigital.dev.br/web/html5-novos-elementos-de-estrutura/> Acessado em: 25 de Ago. 2021.
- 6 ARAÚJO, Eduardo Cruz; SOARES, Luiz Fernando Gomes. **Designing iDTV Applications through Interactive Storyboards**. In: Proceedings of the 20th Brazilian Symposium on Multimedia and the Web. 2014. p. 5-12.
- 7 BARRETO, F. **Uma Proposta de Extensão do Middleware Ginga-NCL para Interação Multimodal e Suporte Multiusuário em Ambientes Hiperfídia**. 2021. Tese de Doutorado, Universidade Federal Fluminense, Niterói.
- 8 BATTISTI, Douglas et al. **A communication middleware for developing serious games that use personal health devices**. Proceedings of SBGames, p. 563-566, 2019.
- 9 BATTLE, Robert; BENSON, Edward. **Bridging the semantic Web and Web 2.0 with representational state transfer (REST)**. Journal of Web Semantics, v. 6, n. 1, p. 61-69, 2008.
- 10 BICHARD, John et al. **Backseat playgrounds: pervasive storytelling in vast location based games**. In: International Conference on Entertainment Computing. Springer, Berlin, Heidelberg, 2006. p. 117-122.
- 11 BOLT, Richard A. “**Put-that-there**” **Voice and gesture at the graphics interface**. In: Proceedings of the 7th annual conference on Computer graphics and interactive techniques. 1980. p. 262-270.
- 12 CASILLO, Mario et al. **A Context-Aware Approach for a Collaborative, Pervasive and Adaptive Digital Storytelling**. In: DMS. 2016. p. 165-174.

- 13 DE OLIVEIRA RODRIGUES, Renato et al. **A Proposal for Supporting Sensory Effect Rendering in Ginga-NCL**. In: Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web. SBC, 2019. p. 187-189.
- 14 DEAL. 2019. **IoT – MQTT e REST, os protocolos utilizados no mundo IoT. O MQTT E O REST São Os Protocolos Mais Utilizados No Mundo Do IoT E Possuem Diversos Benefícios E Pontos Positivos**. Disponível em: <https://www.deal.com.br/blog/iot-mqtt-e-rest-os-protocolos-utilizados-no-mundo-iot/>. Acesso em: 5 nov. 2021.
- 15 DE OLIVEIRA RODRIGUES, Renato; DOS SANTOS, Joel André Ferreira. **EFEITOS SENSORIAIS EM NCL**. 2019.
- 16 DURIE, Bruce. 2017. **Senses Special: Doors of Perception**. Disponível em: <https://www.newscientist.com/article/mg18524841-600-senses-special-doors-of-perception>. Acessado em: 11 jan. 2021.
- 17 FREESZ JR, Marco A.; YUNG, Ludmila; MORENO, Marcelo. **STorM: A Hypermedia Authoring Model for Interactive Digital Out-of-Home Media**. In: Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web. 2017. p. 41-48.
- 18 FREESZ JR, Marco A. **STorM: Um Modelo de Autoria Hipermissão para Out-of-Home**. Tese (Mestrado em Ciência da Computação) – Universidade Federal de Juiz de Fora. Juiz de Fora. 2017.
- 19 GUEDES, Alan LV et al. **Extending ncl to support multiuser and multimodal interactions**. In: Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web. 2016. p. 39-46.
- 20 GUEDES, Alan Lívio Vasconcelos; DE ALBUQUERQUE AZEVEDO, Roberto Gerson; BARBOSA, Simone Diniz Junqueira. **Extending multimedia languages to support multimodal user interactions**. Multimedia tools and applications, v. 76, n. 4, p. 5691-5720, 2017.
- 21 GUEDES, Alan LV et al. **Modeling Multimodal-Multiuser Interactions in Declarative Multimedia Languages**. In: Proceedings of the ACM Symposium on Document Engineering 2019. 2019. p. 1-10.
- 22 ISO/IEC 23005-1:2020, 2020. **Information technology — Media context and control — Part 1: Architecture**. International Organization for Standardization, Geneva, CH.
- 23 ISO/IEC 23005-2:2018, 2018. **Information technology — Media context and control — Part 2: Control information. Standard**. International Organization for Standardization, Geneva, CH.
- 24 ISO/IEC 23005-3:2019, 2019. **Information technology — Media context and control — Part 3: Sensory information. Standard**. International Organization for Standardization, Geneva, CH.

- 25 ITU-T 2014. **Nested context language (NCL) and Ginga-NCL for IPTV services**. Recommendation ITU-T H.761. <http://www.itu.int/rec/T-REC-H.761>
- 26 JEWELL, Michael O.; HOOPER, Clare J. **Linking locations: Storytelling with pervasive technology**. 2011.
- 27 JOSUÉ, Marina et al. **Modeling sensory effects as first-class entities in multimedia applications**. In: Proceedings of the 9th ACM Multimedia Systems Conference. 2018. p. 225-236.
- 28 JOSUÉ, M. I. P. **Preparação de objetos de mídia e efeitos sensoriais para formatação de documentos mulsemídia**. 2021. Tese de Doutorado, Universidade Federal Fluminense, Niterói.
- 29 JUNQUEIRA, Bruna Dalmaso. **Competência emocional e competência informacional: a aquisição qualificada de competências a partir da narração de histórias**. 2010.
- 30 LEITE, JR Emiliano; MARTINS, Paulo S.; URSINI, Edson L. **A Internet das Coisas (IoT): Tecnologias e Aplicações**. no. December, 2017.
- 31 LUCID. [n.d.]. **Where teams come together to see and build the future**. Disponível em <https://lucid.com>. Acesso em: 6 de jun. 2021
- 32 MATTOS, Douglas P.; MUCHALUAT-SAADE, Débora C. **MultiSEM: A mulsemedia model for supporting the development of authoring tools**. In: Proceedings of the 24th Brazilian Symposium on Multimedia and the Web. 2018. p. 109-116.
- 33 MATTOS, Douglas Paulo de; MUCHALUAT SAADE, Débora Christina. **STEVE: Spatial-Temporal View Editor for Authoring Hypermedia Documents**. In: Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web. 2016. p. 63-70.
- 34 MONTEVECCHI, Eyre Brasil B.; SAADE, Débora CM. **Efeitos Sensoriais em Aplicações Multimídia**.
- 35 MORAES, DANIEL. **Descubra o que é Web Scraping e como isso pode ajudar a sua estratégia digital**. 2018. Disponível em: <https://rockcontent.com/br/blog/web-scraping/>. Acesso em: 6 nov. 2021.
- 36 MOZILLA. **Javascript**. 2021. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em: 6 nov. 2021.
- 37 MUCHALUAT, D. C. **Browsers e Trilhas para Documentos Hipermídia Baseados em Modelos com Composições Aninhadas**. 1996. Tese de Doutorado. Dissertação de Mestrado, Departamento de Informática, PUC-Rio.
- 38 MURRAY, Janet Horowitz. **Hamlet on the holodeck: The future of narrative in cyberspace**. MIT press, 1997.

- 39 OASIS. **MQTT Version 5.0**. Disponível em: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html> . Acesso em: 5 de novembro de 2021.
- 40 NERI, Renan; LOMBA Matheus; BULHÕES Gabriel. **MQTT**. Disponível em: <https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/mqtt/#comosurgiu>. Acesso em: 5 de novembro de 2021.
- 41 NIAN TIC. [n.d.]. **Pokemon Go**. Acessado 4 de Junho de 2021 de <https://pokemongolive.com/en/>
- 42 NIGAY, Laurence; COUTAZ, Joëlle. **Multifeature systems: The care properties and their impact on software design**. Intelligence and multimodality in multimedia interfaces, 1997.
- 43 OLIVEIRA, L.; DOS SANTOS, S. DE LIMA VIEIRA; CARVALHO, S. T. **Salus ciber ludens: jogo aplicado para o autocuidado em diabetes em uma experiência conectante**. SBGA MES, 2020.
- 44 PLØHN, Trygve; LOUCHART, Sandy; AALBERG, Trond. **Dynamic pervasive storytelling in long lasting learning games**. 2015.
- 45 POPAI BRASIL. **Point of Purchase Advertising International**. Disponível em: <http://www.popaibrasil.com.br> Acessado em: 2 ago. 2021.
- 46 PRATES, N. et al. **Ameaças de segurança, defesas e análise de dados em IoT baseada em SDN**. Minicursos SBSeg, p. 1-50, 2018.
- 47 SERIQUE, Raissa. 2019. **O QUE é a tendência de edge computing e qual a sua relevância no contexto atual?** isponível em: rockcontent.com/br/blog/edge-computing/. Acesso em: 5 nov. 2021.
- 48 PUC-RIO, TeleMidia Lab, n.d. **Exemplo 1 - Modelo de Execução**. [online] [Telemedia.puc-rio.br](http://www.telemedia.puc-rio.br). Disponível em: http://www.telemedia.puc-rio.br/francisco/nclua/tutorial/exemplo_01.html. Acessado em: 18 fev. 2021.
- 49 QUANTIC DREAM. **Detroit: Become Human**. ID da build na Steam 5354800. [Paris]: Quantic Dream, 2020.
- 50 RICHARDS, Robert. **Representational state transfer (rest)**. In: Pro PHP XML and web services. Apress, Berkeley, CA, 2006. p. 633-672.
- 51 SAHA, Debashis; MUKHERJEE, Amitava. **Pervasive computing: a paradigm for the 21st century**. *Computer*, v. 36, n. 3, p. 25-31, 2003.
- 52 SANTAELLA, Lucia et al. **Desvelando a Internet das coisas**. 2018.
- 53 SHI, Weisong et al. **Edge computing: Vision and challenges**. IEEE internet of things journal, v. 3, n. 5, p. 637-646, 2016.
- 54 SOARES, Luis. F. G.; BARBOSA, SDJ **Programando em NCL 3.0: desenvolvimento de aplicações para middleware Ginga: TV digital e Web**. Editora Campus 2 (2012), 549.

- 55 SOARES, Luiz Fernando Gomes; RODRIGUES, Rogério Ferreira. **Nested context model 3.0: Part 1-ncm core**. Relatório Técnico de Pesquisa da série de Monografias do Departamento de Informática da PUC-Rio, v. 12, 2005.
- 56 SOARES, Luiz Fernando Gomes et al. **Multiple exhibition devices in DTV systems**. In: Proceedings of the 17th ACM international conference on Multimedia. 2009. p. 281-290.
- 57 SOUSA, Flávio RC; MOREIRA, Leonardo O.; MACHADO, Javam C. **Computação em nuvem: Conceitos, tecnologias, aplicações e desafios**. II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI), p. 150-175, 2009.
- 58 TIMMERER, Christian et al. **Interfacing with virtual worlds**. Network and Electronic Media Summit, 2009.
- 59 TOLEDO, Luciano Augusto; DE FARIAS SHIAISHI, Guilherme. **Estudo de caso em pesquisas exploratórias qualitativas: um ensaio para a proposta de protocolo do estudo de caso**. Revista da FAE, v. 12, n. 1, 2009.
- 60 TURNER. [n.d.]. **Gumball Medieval**. Disponível em: <https://www.amazon.com.br/Turner-Gumball-Medieval/dp/B07YNV33LP>. Acessado em: 6 jun. 2021
- 61 TURNER. [n.d.]. **Trivia Ursos Sem Curso**. Disponível em: <https://www.amazon.com.br/Turner-Trivia-Ursos-Sem-Curso/dp/B07XC33ZYM>. Acessado em: 6 jun. 2021
- 62 W3C. **HTML5 Reference The Syntax, Vocabulary and APIs of HTML5** Disponível em: <https://dev.w3.org/html5/html-author/#basic-templates>. Acessado em: 20 out. 2020.
- 63 W3C. **Cascading Style Sheets Basic User Interface Model Level 3**. 2012. Disponível em: <http://www.w3.org/TR/css3-ui/>. Acessado em: 26 nov. 2020.
- 64 W3SCHOOLS.**HTML Canvas Graphics**. Disponível https://www.w3schools.com/html/html5_canvas.asp. Acessado em: 23 out. 2020.
- 65 W3C. **Synchronized Multimedia Integration Language (SMIL 3.0)**. December 2008. Disponível em: <http://www.w3.org/TR/REC-smil/>. Acesso em: 26 nov. 2020.
- 66 YAMIN, Adenauer Corrêa. **Arquitetura para um ambiente de grade computacional direcionado às aplicações distribuídas, móveis e conscientes do contexto da computação pervasiva**. 2004.

A Fragmento de Lux Ferre

Fragmento de Lux Ferre representado no formato JSON definido para leitura da máquina de orquestração.

Listing A.1 – Fragmento modificado de Parte da narrativa Lux Ferre em JSON

```

1 {
2   "pst": {
3     "$": {
4       "id": "idDoElemento",
5       "name": "Nome do Elemento",
6       "title": "Titulo da Obra",
7       "circunstancias": {
8         "to0ccur": {"interator": [{ "age": 18}]}
9       },
10      "fatherID": "none",
11      "status": "enabled"
12    },
13    "episode": [
14      {
15        "$": {
16          "id": "primeiroEpisodio",
17          "name": "PrimeiroEpisodio",
18          "title": "Primeiro Episodio, Lux Ferre",
19          "circunstancias": {
20            "to0ccur": "true"
21          },
22          "status": "enabled"
23        },
24        "experience": [
25          {
26            "$": {
27              "id": "exp1",
28              "name": "Acordando",
29              "circunstancias": {
30                "to0ccur": {"devices": [{ "type": "headphones"}]}
31              },
32              "status": "enabled"
33            },
34            "sequence": [
35              {
36                "$": {
37                  "name": "sequenciaInicial",
38                  "id": "sequenciaInicialexp1",
39                  "status": "enabled",
40                  "circunstancias": {

```

```

41         "toOccur": "true"
42     },
43 },
44 "media": [
45     {
46         "$": {
47             "id": "audioacordando",
48             "type": "audio",
49             "src": "acordando.mp3",
50             "status": "enabled",
51             "prev": "",
52             "next": "sensorialEffect1exp1"
53         }
54     }
55 ],
56 "sensorialeffect": [
57     {
58         "$": {
59             "id": "audioacordando",
60             "type": "temperature",
61             "properties": "{celcius:18, delay:180, dur:30}",
62             "status": "enabled",
63             "prev": "",
64             "next": "dp1exp1"
65         }
66     }
67 ],
68 "decisionpoint": [
69     {
70         "$": {
71             "timeout": "500",
72             "status": "enabled",
73             "id": "dp1exp1",
74             "timeOutAction": "{seek:#audioDentroDoPredio, stop:this}",
75             "prev": "sensorialEffect1exp1",
76             "next": ""
77         },
78         "option": [
79             {
80                 "$": {
81                     "id": "op1exp1",
82                     "name": "primeiraEscolha",
83                     "condition": "voiceRecognition(['lux', 'primeiro'])",
84                     "action": "{enable:#SequenciaLuxAto2,

```

```
85         disable:#SequenciaFerreAto2, save:aura.fe
86         =10}"
87     },
88     {
89         "$": {
90             "id": "op2exp1",
91             "name": "segundaEscolha",
92             "condition": "voiceRecognition(['ferre',
93                 'segundo','sair'])",
94             "action": "{enable:#SequenciaLuxAto2,
95                 disable:#SequenciaLuxAto2, save:aura.fe=-
96                 10}"
97         }
98     }
99 ]
100 }
101 ]
102 }
103 ]
104 }
105 }
```

B Formulário de Avaliação da Compreensão do Modelo Pervasivo e Resultados

Neste apêndice encontram-se figuras referentes ao questionário aplicado para a avaliação da compreensão das principais entidades do modelo PST pelos integrantes do LMD, conforme mencionado em 5.1.

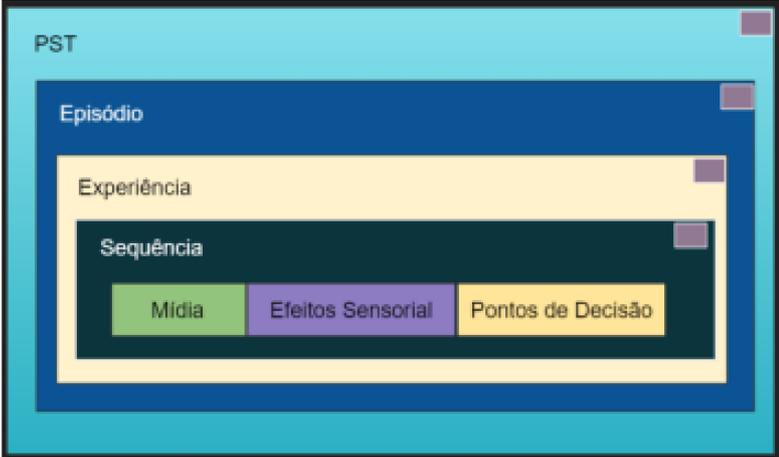
Figura 33 - Avaliação de Compreensão do Modelo Pervasivo
- Parte 1

Avaliação de entendimento do Modelo Pervasivo
Formulário para medir o nível de absorção do modelo.

1. Nome

2. Email

Resumo do modelo. (Lembrando que quadro roxo na lateral represente o elemento "Circunstância")



```
graph TD
    PST[PST] --- Episodio[Episódio]
    Episodio --- Experiencia[Experiência]
    Experiencia --- Sequencia[Sequência]
    Sequencia --- Midia[Mídia]
    Sequencia --- Efeitos[Efeitos Sensorial]
    Sequencia --- Pontos[Pontos de Decisão]
```

Fonte: Elaborado pelo autor (2021)

Figura 34 - Avaliação de Compreensão do Modelo Pervasivo
- Parte 2

3. Qual seu nível de entendimento do modelo

Marcar apenas uma oval.

- [1] Não compreendi
- [2] Compreendi um pouco / menor parte
- [3] Compreendi Parcialmente
- [4] Compreendi a maior parte
- [5] Compreendi por completo

4. Acredita que o modelo ajuda na criação de histórias imersivas frente ao que existe atualmente?

Marcar apenas uma oval.

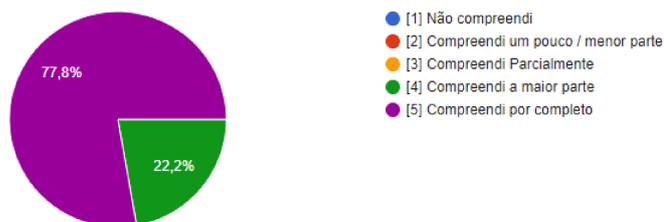
- Sim
- Não
- Outro: _____

Fonte: Elaborado pelo autor (2021)

Figura 35 - Resposta da Avaliação de Compreensão do
Modelo Pervasivo - Parte 1

Qual seu nível de entendimento do modelo

9 respostas

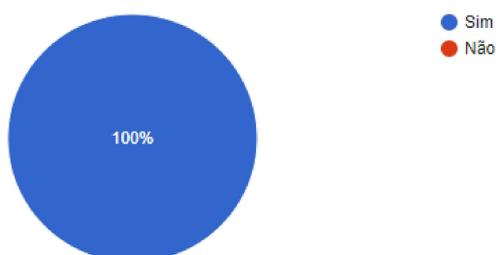


Fonte: Elaborado pelo autor (2021)

Figura 36 - Resposta da Avaliação de Compreensão do Modelo Pervasivo - Parte 2

Acredita que o modelo ajuda na criação de histórias imersivas frente ao que existe atualmente?

9 respostas

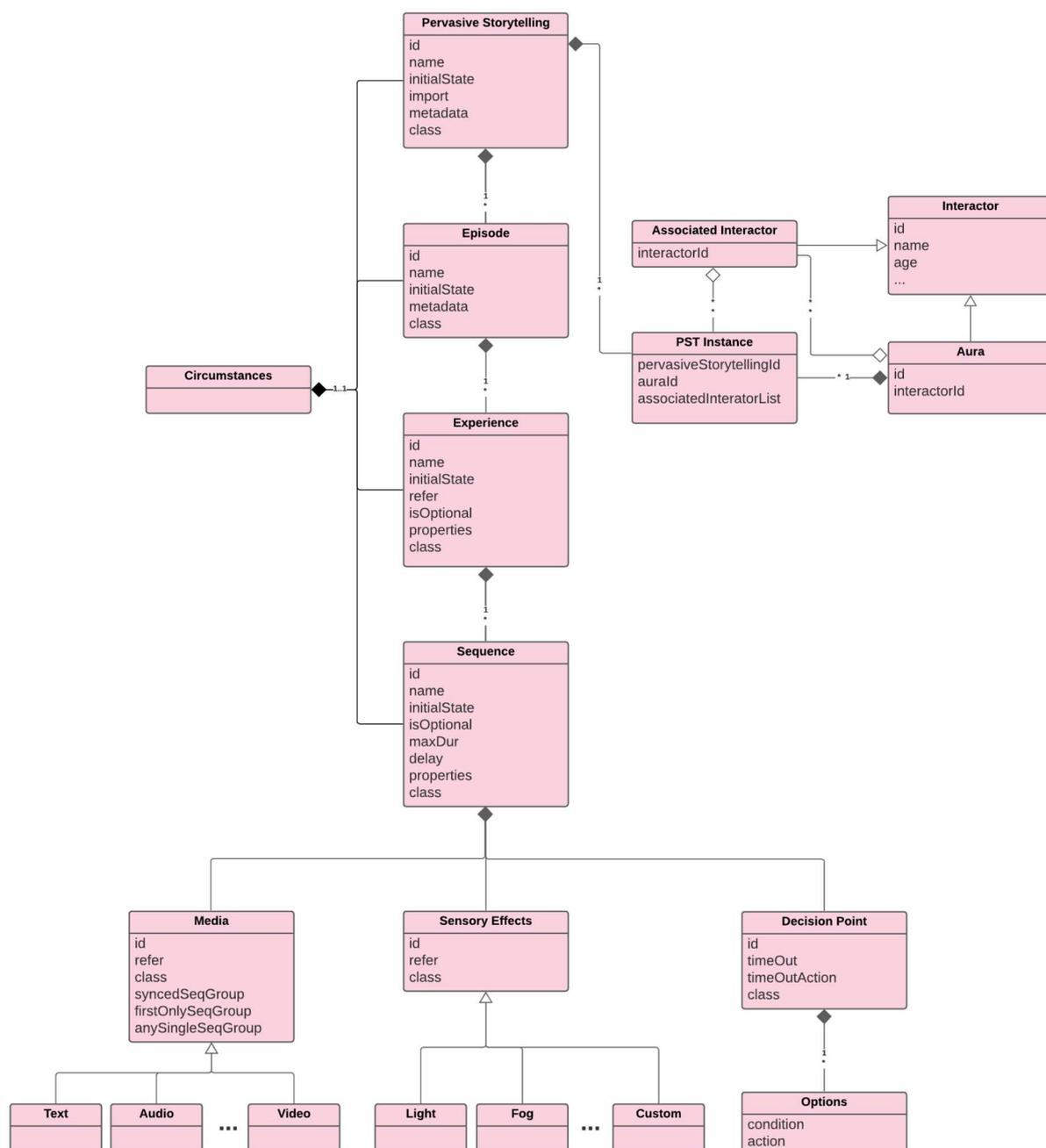


Fonte: Elaborado pelo autor (2021)

C Diagrama de classes do Modelo PervasiveStorytelling

Neste apêndice encontra-se um diagrama de classes do modelo PST para ser tomada como referência.

Figura 37 - Diagrama de classes do Modelo PST

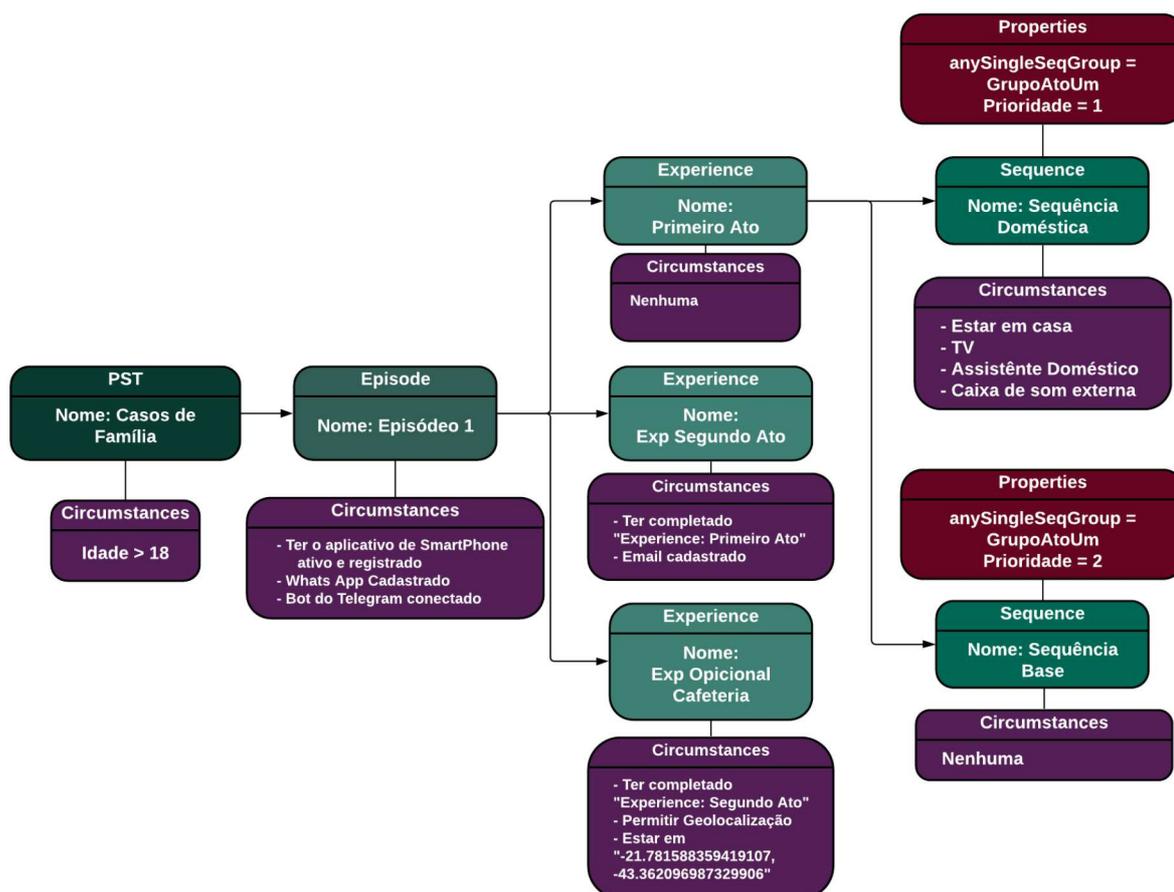


Fonte: Elaborado pelo autor (2021)

D Representação da narrativa de Anna seguindo o Modelo PervasiveStorytelling

Neste apêndice é apresentado um fragmento reduzido e simplificado da narrativa de Anna apresentada na subseção 2.3 nas figuras 2, 3, 4 e 5. Na figura 38, é criada para a história uma entidade primária de PST com o nome "Casos de Família", referente ao título da história, sendo esta responsável por representar o todo. A PST possui, a princípio, apenas um episódio (*Episode*), denominado "Episódio 1", que exige que o interator possua ao menos um *Smartphone* com o aplicativo específico instalado para uso da história, além de ter registrado um número de contato associado a uma conta de "WhatsApp" e "Telegram" para uso do *chatbot*. Esses requisitos definidos nas *circumstances* são os mínimos necessários para garantir que o interator consiga consumir a história. Caso ele não possua os requisitos mínimos, infelizmente a mesma não poderá iniciar.

Figura 37 - Representação parcial da narrativa de Anna - Parte 1



Fonte: Elaborado pelo autor (2021)

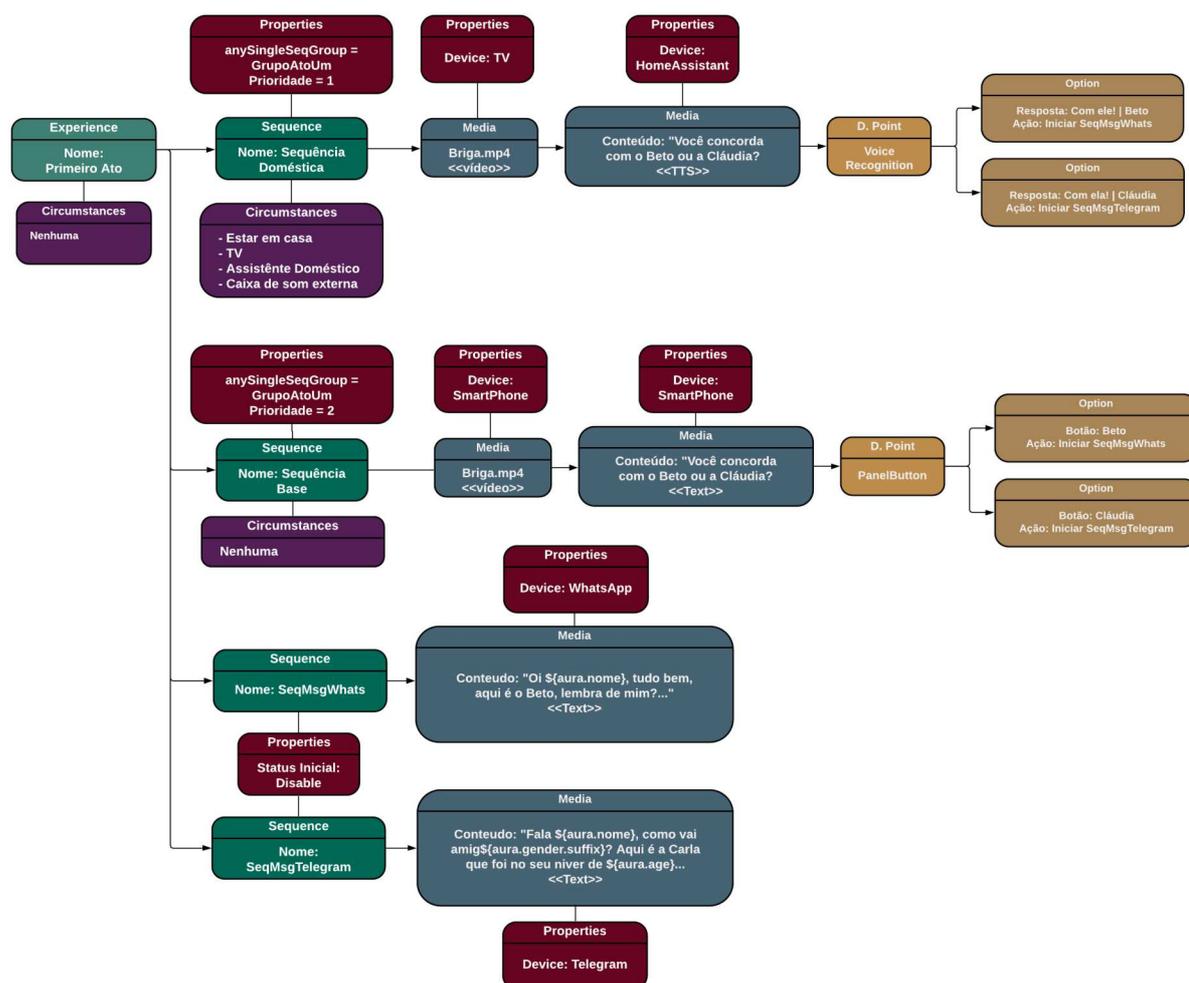
Ao "Episódio 1", são associadas três Experiências (*Experiences*). A primeira experiência é denominada "Primeiro Ato", onde é contada a primeira parte da

história, que pode ser consumida em uma casa com os dispositivos mínimos, ou em um celular independente de onde se encontre. Como o "Primeiro Ato" é a unidade primária definida pelo autor, ou seja, a primeira parte da história a ser contada, o mesmo define não haver circunstâncias mínimas para que ela se inicie, sendo a primeira a ser executada em um caso de apresentação. As demais experiências possuem então circunstâncias relacionadas ao término das demais experiências do episódio para garantir ordem. No caso da experiência "Segundo Ato", ela necessita que o interator já tenha consumido a experiência "Primeiro Ato" e possua um *email* cadastrado em sua aura. A experiência "Exp Opcional Cafeteria" necessita que o interator tenha consumido por completo a experiência "Segundo Ato", possibilite acesso à geolocalização, e se encontre em um ponto específico no mapa.

A experiência "Primeiro Ato", conforme pode ser visto nas figuras 2, 3, possuem duas formas de serem contadas, que variam conforme o meio em que se encontra o interator. Caso o interator esteja em casa e possua todos os dispositivos necessários, a história se desenrolará pela sequência (*sequence*) "Sequência Doméstica". Caso contrário, ocorrerá pela "Sequência Base". A sequência base não define em suas circunstâncias requisitos mínimos para sua execução, dado que utiliza de todos os recursos e dispositivos solicitados para a execução da história nas circunstâncias do Episódio (Episódio 1). Para garantir que, caso o interator possua as circunstâncias para ser executada a "Sequência Doméstica", ela seja executada preferencialmente, e jamais ocorra as duas sequências simultaneamente, tanto a sequência "Sequência Doméstica" quanto "Sequência Base" foram associadas a um *anySingleSeqGroup* em comum nomeado "GrupoAtoUm". Com isso, o autor determina que apenas uma sequência que estiver nesse grupo será executada por vez, e através da definição de nível de prioridade, o autor sinaliza para que a preferência de execução, sempre seja da "Sequência Doméstica". Quando uma das duas sequências for finalizada, a experiência "Primeiro Ato" poderá então ser completada, dando início à experiência "Segundo Ato".

Como dito, a Experiência "Primeiro Ato" possui mais de uma sequência para contar um mesmo fragmento da história, mas em contextos diferentes, conforme demonstrando na figura 39. A sequência "Sequência Base", por não possuir nenhum tipo de restrição ou uso de dispositivo específico, utiliza única e especificamente do "SmartPhone", sendo esse um dispositivo obrigatório para consumo do Episódio. Todas as mídias são definidas para serem executadas no dispositivo, e o ponto de decisão apresenta um painel com as opções na tela do interator na hora da tomada de decisão. Já na "Sequência Doméstica", devido à existência de outros dispositivos, é definido pelo autor que o vídeo seja reproduzido em uma TV, a fala anterior à tomada de decisão falada por *Text-To-Speech* (TTS) por uma assistente doméstica, e a entrada da decisão obtida por voz. Apesar das diferenças, ambos os pontos de decisão das duas sequências resultam na liberação ou não de uma das sequências "SeqMsgWhats" ou "SeqMsgTelegram", que disparam respectivamente as mensagens via "WhatsApp" ou "Telegram", obtendo um mesmo resultado.

Figura 37 - Representação parcial da narrativa de Anna - Parte 2



Fonte: Elaborado pelo autor (2021)