

Hedvan Fernandes Pinto

**Autoria de *e-books* Interativos: Modelo
Conceitual Fábulas e Requisitos.**

São Luís

2017

Hedvan Fernandes Pinto

**Autoria de *e-books* Interativos: Modelo Conceitual
Fábulas e Requisitos.**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Orientador: Carlos de Salles Soares Neto

São Luís

2017

Hedvan Fernandes Pinto

Autoria de *e-books* Interativos: Modelo Conceitual Fábulas e Requisitos./ Hedvan Fernandes Pinto. – São Luís, 2017-

78p. : il. (algumas color.) ; 30 cm.

Orientador: Carlos de Salles Soares Neto

Dissertação de mestrado – Universidade Federal do Maranhão

Centro de Ciências Exatas e Tecnologia

Programa de Pós-Graduação em Ciência da Computação, 2017.

1. Interactive eBook. 2. Multimedia Authoring. 3. Conceptual Model. 4. Domain specific modeling language. 5. Storytelling. I. Carlos de Salles Soares Neto. II. Universidade Federal do Maranhão. IV. Título

Este trabalho é dedicado à minha família e amigos.

Agradecimentos

Agradeço à minha família e a todos que direta ou indiretamente guiaram os meus passos até este momento. Agradeço e dedico à minha mãe e minha avó, pessoas a quem admiro e amo bastante principalmente por ser sempre companheira, amiga, conselheira, carinhosa e atenciosa.

Agradeço a todos da minha família e amigos que sempre me encorajaram a continuar tentando e a nunca desistir. Meus companheiros de Laboratório com os quais compartilho felicidades e dificuldades.

Agradeço aos grandes amigos que sempre estiveram lá para me ouvir reclamar e sempre acreditaram em mim mais que eu mesmo. Em especial a Wallas, Thiago, Alex, Busson, André, Fábio, Dilson, Daniel, Ruy, Rômulo, Felipe, e tantos outros que seria demais para citar por nome aqui, sem o apoio de vocês eu não chegaria tão longe. Obrigado por me apoiarem.

Agradeço também a todos os professores do Departamento de Informática da UFMA. Desses dez anos eu só lamento por não ter aproveitado mais do conhecimento que tinham a oferecer, tanto profissional e acadêmico quanto de suas experiências de vida. Sou grato a vocês pelo que sou hoje, e posso afirmar com orgulho que fui formado por vocês.

Agradecimento especial ao meu orientador, Carlos de Salles, com quem tive a honra e o prazer de trabalhar. Que acreditou em meu trabalho, que acreditou em mim, me aguentou todos esses anos e possibilitou a concretização deste. Sem o seu apoio e conselhos não seria possível chegar onde estou.

Agradecimento especial também ao Roberto Gerson cuja participação neste trabalho foi fundamental. Muito obrigado pela paciência, por toda as dicas, atenção aos detalhes, e críticas para melhoria do trabalho. Este trabalho não seria metade do que é hoje sem sua ajuda. Você me mostrou como um real profissional trabalha, um exemplo que desejo seguir.

Professor Carlos Salles, e Roberto Gerson espero que possamos trabalhar novamente juntos no futuro.

*“The greatest mistake you can make in life
is to continually fear you will make one.”
(Elbert Hubbard)*

Resumo

Hoje em dia, os *tablets* e *smartphones* são comumente usados por crianças tanto para entretenimento quanto para educação. *E-books* interativos em dispositivos móveis permitem uma experiência mais rica quando comparada com livros tradicionais, sendo potencialmente mais envolvente e divertido para os leitores. Contudo, para explorar os recursos mais interessantes nesses ambientes, os autores têm pouco suporte. No sentido de que não existem muitas ferramentas de apoio de nível elevado e que estas funcionalidades são normalmente acessíveis apenas através de programação. Neste trabalho, pretendemos extrair as principais características de *e-books* interativos e propor um modelo chamado Fábulas que permite aos autores criarem *e-books* interativos declarativamente. O modelo foi concebido tomando como ponto de partida uma análise sistemática de *e-books* interativos, ferramentas de autoria e linguagens de autoria para aplicações multimídia. Além disso, esta dissertação apresenta um estudo de caso para a implementação de Fábulas para navegadores web com Javascript e HTML e para a linguagem SceneSync.

Palavras-chave: E-book interativo, autoria multimídia, modelo conceitual, linguagem de modelagem de domínio específico, storytelling.

Abstract

Nowadays, tablet and smartphones are commonly used by children for both entertainment and education. Interactive *e-book* for mobile devices offer a enhanced experience when compared to traditional books, being potentially more engaging and fun for readers. However, to explore the most interesting features in these environments, authors have scarce support. In the sense that there are not many high level support tools and that these features are usually only accessible programmatically. In this work, we intend to extract as main characteristics of interactive *e-books* and propose a model called Fábulas that allow authors to create declarative interactive e-books. The model was designed based on a systematic analysis of interactive *e-books*, authoring tools and authoring languages for multimedia applications. In addition, this dissertation presents a case study for an implementation of Fábulas for web browsers with Javascript and HTML and for a SceneSync language.

Keywords: Interactive eBook, Multimedia Authoring, Conceptual Model, domain specific modeling language, storytelling.

Lista de ilustrações

Figura 1 – Formas de uso da plataforma Fábulas de acordo com os diferentes perfis de usuário.	15
Figura 2 – Layout do site <i>Elements of a Story</i>	22
Figura 3 – Dois exemplos de jogos dos exercícios de fixação encontrados no site <i>Elements of a Story</i>	23
Figura 4 – Menu inicial do aplicativo <i>Tales with Gigi</i>	23
Figura 5 – História da <i>Cinderella</i> , com os botões de controle ocultos e com os controles apresentados	24
Figura 6 – Jogos disponíveis no aplicativo <i>Tales with Gigi</i>	25
Figura 7 – Menu inicial do aplicativo <i>Classic Fairy Tales</i> para seleção do idioma	26
Figura 8 – Tela da história <i>Cinderella</i> no aplicativo <i>Classic Fairy Tales</i>	26
Figura 9 – Exemplo de jogos no aplicativo <i>Classic Fairy Tales</i>	27
Figura 10 – menu de seleção dos contos no aplicativo Livro mágico	28
Figura 11 – Um dos desvios onde o usuário pode decidir como a história deve prosseguir	28
Figura 12 – Um dos desvios onde o usuário pode decidir como a história deve prosseguir	29
Figura 13 – Aplicativo <i>iStoryBooks</i>	30
Figura 14 – e-book <i>The Little Mermaid</i>	31
Figura 15 – Interface do sistema <i>Genially</i>	37
Figura 16 – Interface do aplicativo <i>My Story Book Creator School Edition</i>	38
Figura 17 – Interface do aplicativo <i>TyranoBuilder</i>	41
Figura 18 – Modelo de narrativas do Fábulas	46
Figura 19 – Modelo conceitual de um eBook interativo como definido neste trabalho	47
Figura 20 – Exemplo da primeira página do e-book O pequeno cavaleiro.	48
Figura 21 – Demonstração de posicionamento de uma imagem de acordo com os containers.	53
Figura 22 – Exemplo de apresentação de uma cena com camadas definidas.	54
Figura 23 – Exemplo de imagens de uma animação de <i>frame</i>	57
Figura 24 – Máquina de estados dos eventos NCM	58
Figura 25 – Representação de um agente representado uma porta trancada.	60
Figura 26 – Diagrama de navegação entre as páginas do e-book interativo “O pequeno cavaleiro”	66
Figura 27 – Agente “bush” e agente “key” interagindo com o agente “door”.	68
Figura 28 – Cenas da narrativa “Ajude o Pirata”	72

Lista de tabelas

Tabela 1 – Estrutura final proposta pelos participantes da pesquisa sobre os elementos dos <i>e-books</i> infantis interativos	19
Tabela 2 – Características presentes no <i>e-book Elements of a Story</i> - Cinderella . .	22
Tabela 3 – Características presentes no <i>e-book Tales with gigi</i>	25
Tabela 4 – Características presentes no <i>e-book Classic fairy tales</i>	27
Tabela 5 – Características presentes no <i>e-book Livro mágico</i>	28
Tabela 6 – Características presentes no <i>e-book The Jungle Book</i>	29
Tabela 7 – Características presentes no <i>e-book iStoryBooks</i>	30
Tabela 8 – Características presentes no <i>e-book The Little Mermaid</i>	31
Tabela 9 – Requisitos revisados para <i>e-books</i> interativos	32
Tabela 10 – Eventos de interação do player	51
Tabela 11 – Ações suportadas pelo Fábulas	52
Tabela 12 – Equivalências entre máquina de estado de agente e modelo Fábulas . .	59
Tabela 13 – Lista de propriedades do modelo Fábulas	62
Tabela 14 – Elementos do modelo Fábulas implementados ou modificados pela máquina de execução	65
Tabela 15 – Elementos do módulo Interaction	71

Lista de abreviaturas e siglas

e-book	eletronic book
SSM	SceneSync Model
NCL	Nested Context Language
NCM	Nested Context Model
HTML	HyperText Markup Language
PDF	Portable Document Format
CSS	Cascading Style Sheets
W3C	World Wide Web Consortium
XML	Extensible Markup Language
WYSIWYG	What Yout See is What You Get
EPUB	Electronic Publication
SMIL	Synchronized Multimedia Integration Language
TVDi	TV digital interativa
DSL	Domain-Specific Language
MDE	Model-Driven Engineering

Sumário

1	INTRODUÇÃO	14
2	REQUISITOS PARA <i>E-BOOKS</i> INFANTIS INTERATIVOS	17
2.1	Padrões de interação para <i>e-books</i> infantis interativos	17
2.2	Catálogo de livros	20
2.2.1	<i>Elements of a Story - Cinderella</i>	21
2.2.2	<i>Tales with gigi</i>	22
2.2.3	<i>Classic fairy tales</i>	25
2.2.4	Livro mágico	27
2.2.5	<i>The Jungle Book</i>	28
2.2.6	<i>iStoryBooks</i>	29
2.2.7	<i>The Little Mermaid</i>	30
2.3	Características de <i>e-books</i> interativos	31
2.4	Considerações finais	31
3	TECNOLOGIAS RELACIONADAS	33
3.1	Ferramentas de autoria	34
3.1.1	<i>iBooks Author</i>	35
3.1.2	<i>BlueGriffon EPUB Edition</i>	36
3.1.3	CircularFLO	36
3.1.4	TaleSpring	37
3.1.5	Genially	37
3.1.6	<i>My Story Book Creator School Edition</i>	37
3.1.7	Kitaboo	38
3.1.8	Kwik	38
3.1.9	Demibooks	39
3.1.10	Kindle Textbook Creator	40
3.1.11	Kindle Comic Creator	40
3.1.12	Kindle Kids' Book Creator	40
3.1.13	TyranoBuilder	41
3.2	Linguagem de autoria	41
3.2.1	EPUB	42
3.2.2	HTML5	43
3.2.3	SMIL	43
3.2.4	NCL	44
3.2.5	SceneSync	44

4	MODELO CONCEITUAL MULTIMÍDIA FÁBULAS	46
4.1	Interação do Usuário: Causalidade e Sincronismo	50
4.1.1	Eventos	50
4.1.2	Ações	51
4.2	Mídias	52
4.2.1	Imagem	52
4.2.2	Texto	53
4.2.3	Vídeo	54
4.2.4	Áudio	55
4.2.5	Animação	57
4.2.5.1	Animação de frame	57
4.2.5.2	Transições	58
4.3	Agentes	58
4.4	Propriedades	60
5	MODELANDO NARRATIVAS COM O MODELO FÁBULAS	64
5.1	Integração com Web Browsers	64
5.1.1	Exemplo de <i>e-book</i> : O pequeno cavaleiro	66
5.2	SceneSync	71
6	CONCLUSÕES	74
6.1	Trabalhos Futuros	74
6.2	Trabalhos Publicados	75
	REFERÊNCIAS	76

1 Introdução

O primeiro *e-book*, que era chamado na ocasião de *eText*, foi criado em julho de 1971 por Michael Hart no *Project Gutenberg* (HART, 1971). Esse projeto tinha por objetivo criar versões eletrônicas de obras literárias e disseminá-las mundialmente. Lebert (2009) narra o passo a passo da jornada para os *e-books* chegarem ao seu estado atual. Alguns pontos a serem citados são: popularização da internet com a criação da web, adoção da distribuição virtual de livros por algumas editoras, criação de *e-readers*, popularização de tablets etc.

Outro fator a ser destacado é que a popularização dos *e-books* atraiu a atenção de pesquisadores e entusiastas. Isso por sua vez possibilitou o desenvolvimento de novas funcionalidades e ferramentas que, aumentaram mais ainda a sua popularidade. Hoje, os e-books deixaram de ser somente cópias digitais dos livros impressos e podem ser enriquecidos com conteúdos multimídia e interatividade (BENEDETTI, 2016).

Essa evolução, constante desenvolvimento e experimentação com os *e-books* propiciou a criação de diferentes formatos para sua representação. Os principais formatos atuais são PDF (INCORPORATED, 2006b), EPUB (CONBOY et al., 2011) e Mobi (MOBI, 2016). Cada um desses formatos tem vantagens e desvantagens. Por exemplo, o PDF foi projetado para garantir a fidelidade do leiaute das páginas semelhante ou publicações impressas. Por outro lado, o EPUB e o Mobi possibilitam a criação de um leiaute fluido e adaptável para diferentes tamanhos de tela e dispositivos, facilitando a leitura em dispositivos de telas pequenas como *tablets* e *smatphones*.

Tablets e smartphones são plataformas presentes no dia a dia de todos nós e a maioria das pessoas usam essas plataformas para ler *e-books* quando em trânsito para o trabalho ou escola. Esses dispositivos fornecem recursos interativos como tela sensível ao toque, giroscópio, acelerômetro, microfone etc. Existem vários exemplos de *e-books* interativos para estas plataformas que incorporam o conteúdo dos livros em aplicações nativas, geralmente voltadas para o público infantil, como *Alice for iPad*¹, *The Little Mermaid*², *Pinocchio*³ etc. Além dos exemplos citados pode-se encontrar muitos outros navegando nas lojas de aplicativos para mobile Google Play⁴ e iTunes(SHEET, 2001).

Entretanto, para a autoria desses *e-books* interativos é preciso transpor as intenções do autor para uma linguagem interpretável pela máquina. Assim a criação desses *e-books* exige conhecimento profundo de programação e geralmente envolve linguagens

¹ <<https://itunes.apple.com/br/app/alice-for-the-ipad/id354537426>>

² <<https://play.google.com/store/apps/details?id=com.storytoys.TheLittleMermaid.GooglePlay>>

³ <<https://play.google.com/store/apps/details?id=com.tabtale.pinocchikidsstorybook>>

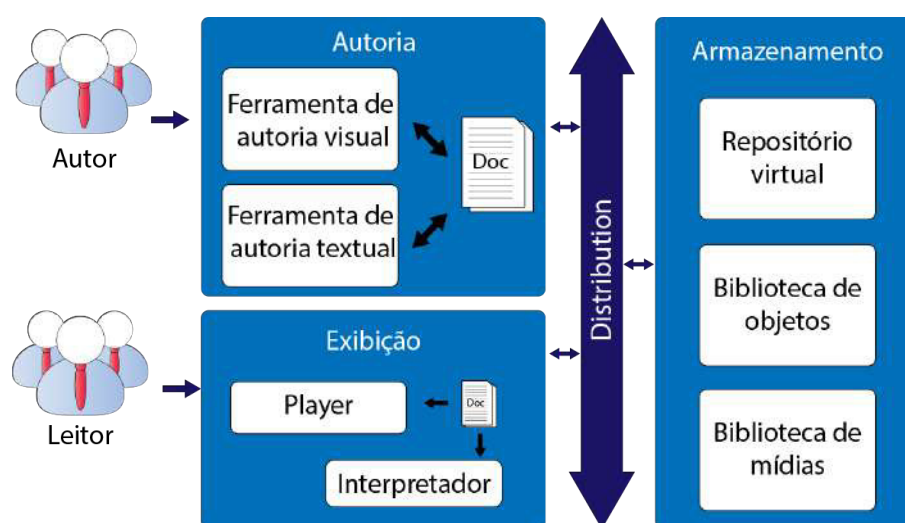
⁴ <<https://play.google.com/store>>

de programação e ferramentas que são complexas demais para o usuário padrão. Isso principalmente porque a maioria dos autores de *e-books* não possui experiência com programação de computadores e lógica de programação. Essa dificuldade força os autores contratarem um programador ou formar equipes multidisciplinares para poder concretizar suas ideias, o que acaba gerando outros problemas de gerenciamento de equipe e falhas de comunicação. Outra opção seria o uso de ferramentas de autoria que auxiliem os autores leigos a criarem seus próprios projetos.

Para criar essas ferramentas é necessário entender o domínio da aplicação para facilitar a comunicação com o usuário. Após a definição do domínio, é necessário formalizar uma linguagem de domínio específico (DSL), nesse caso para o contexto de *e-books* interativos. Segundo [Deursen, Klint e Visser \(2000\)](#) DSLs são definidas como linguagens pequenas e geralmente declarativas que oferecem poder expressivo focado em um domínio de problema particular. Isso as tornam mais simples de serem utilizadas, já que grande parte do conhecimento do domínio já está implícito na própria estrutura da linguagem. Com base na DSL são definidas abstrações, que podem ser gráficas ou não, para a ferramenta de autoria.

Como solução para esse problema é proposta uma plataforma que forneça suporte à todas as fases de desenvolvimento de *e-books* interativos, a plataforma Fábulas. O Fábulas é uma plataforma para dar suporte à autoria de *e-books* interativos e engloba as três fases do desenvolvimento: *autoria*, *exibição*, *distribuição* e *armazenamento*. A Figura 1 apresenta como é a interação dos diferentes elementos da plataforma com os seus usuários.

Figura 1 – Formas de uso da plataforma Fábulas de acordo com os diferentes perfis de usuário.



fonte: elaborado pelo autor

A plataforma Fábulas apresenta dois perfis principais de usuários que são o usuário Leitor e o usuário Autor. O usuário Leitor usa a máquina de execução do Fábulas para

buscar os *e-books* e executá-los. O usuário Autor usa a ferramenta de autoria gráfica para criar os *e-books* interativos.

Esta dissertação se propõe a apresentar o modelo conceitual multimídia **Fábulas**, fundamentado no estudo do domínio da criação de *e-books* interativos. O modelo conceitual do **Fábulas** foca principalmente na criação do documento base para a plataforma. Esse modelo conceitual deve contemplar os principais recursos que um autor de *e-books* interativos necessita discutidos no Capítulo 2.

Como objetivos específicos têm-se:

- O levantamento e classificação das características dos *e-books* interativos;
- Análise de ferramentas de autoria para *e-books* interativos;
- Análise de linguagens de autoria multimídia.

O restante deste trabalho está dividido do seguinte modo. No Capítulo 2 é apresentado o levantamento de requisitos para a definição do modelo de *e-books* interativos. No Capítulo 3 são apresentados formatos utilizados no mercado e ferramentas de autoria para *e-books*. No Capítulo 4 é apresentado o modelo **Fábulas**. O Capítulo 5 mostra a aplicação do modelo **Fábulas** em duas linguagens de autoria multimídia diferentes: HTML5(HICKSON et al., 2014) e SceneSync(BUSSON et al., 2016; BUSSON, 2015). Por fim, o Capítulo 6 discute o trabalho e apresenta as considerações finais.

2 Requisitos para *e-books* Infantis Interativos

Como aponta [Ebner et al. \(2016\)](#), uma definição única para o termo *e-book* não é uma tarefa simples. Isso pode ser atribuído ao fato de que com o tempo os *e-books* evoluíram e ganharam novos atributos e funcionalidades. Atualmente, *e-book* pode ser um arquivo textual simples ou um aplicação multimídia complexa, enriquecida com áudio, vídeo, animação, jogos interativos e até conexão com redes sociais. Esses *e-books* enriquecidos com interatividade e multimídia são chamados de *e-books* interativos (*interactive e-books*) ou *e-books* enriquecidos (*enhanced e-books*) ([MANA et al., 2013](#)).

Os dispositivos móveis (*tablets* e *smartphones*) ajudaram com a popularização dos *e-books* interativos para crianças. Estes *e-books* interativos para crianças (ou *ebooks* infantis interativos) não têm como único foco o entretenimento, mas também têm o propósito de estimular o interesse pela leitura. Este capítulo visa definir as características dos *e-books* infantis interativos.

Para isso, este capítulo está dividido da forma a seguir. Na Seção 2.1 são discutidos os resultados do levantamento de requisitos realizado por um outro trabalho de mestrado ([CRUZ, 2016](#)), o qual também se insere no escopo do mesmo projeto desta dissertação. Na Seção 2.2 apresenta uma lista de *e-books* infantis interativos com o detalhamento de suas funcionalidades, os quais serviram como base para levantamento de novos requisitos. Na Seção 2.4 os requisitos encontrados são apresentados juntamente com as considerações finais do capítulo.

2.1 Padrões de interação para *e-books* infantis interativos

A plataforma Fábulas é um projeto do laboratório TeleMídia Maranhão, do qual esta dissertação trata da engenharia de documentos e do modelo da linguagem de autoria para esses *e-books*. Outras pesquisas já foram realizados sobre a plataforma. Esta dissertação toma como ponto de partida o trabalho de mestrado intitulado “Padrões de Interação para Ferramentas de Autoria de *Storytellings* Infantis” ([CRUZ, 2016](#)), que é discutido nesta seção.

[Cruz \(2016\)](#) realizou um levantamento de requisitos e catalogação de aplicativos de *e-books* focando nas plataformas de dispositivos móveis IOS¹ e Android². Seus objetivos foram:

¹ <https://www.apple.com/br/ios/ios-10/>

² <https://developer.android.com>

- identificar padrões de interação dos usuários com os *e-books* infantis interativos;
- fornecer sugestões para os autores de *e-books* infantis interativos para dispositivos móveis sobre as boas práticas na aplicação desses padrões de interação;
- levantar requisitos para auxiliar o desenvolvimento de ferramentas de autoria para *e-books* infantis interativos.

Embora todos esses pontos se mostrem importantes, pelo tema desta dissertação dá-se foco para os requisitos e padrões de interação levantados no trabalho.

O público alvo para o levantamento de requisitos foi de desenvolvedores de conteúdo e desenvolvedores de *software* no ramo da educação infantil (cientistas da computação, designers, programadores, jornalistas, produtores de marketing etc). As técnicas de investigação usadas para o levantamento de requisitos foram: reuso de requisitos, grupo focal e *card sorting*. Foram realizados três testes utilizando essas técnicas e, de acordo com a opinião dos participantes da pesquisa, foi construída a estrutura de categorias e subcategorias apresentada na Tabela 1.

Segundo a estrutura definida por Cruz, tem-se as seguintes categorias de padrões interativos:

- Ação: representa as categorias de efeitos, animações e ações;
- Gráfico: relativo a presença de ilustrações, fotos, e efeitos diversos;
- Áudio: relativo à música, efeitos sonoros, etc;
- Configuração: menus que alteram características da apresentação da história ou do próprio dispositivo;
- Navegação: relativo à navegação entre as páginas da história;
- Monetização: relativo à forma de captação de renda do desenvolvedor;
- Jogos: jogos e atividades adicionais para tornar a história mais cativante.

Com base nessa estrutura é necessário fazer algumas considerações, levando em conta o ponto de vista de desenvolvedores de *software*. A categoria Monetização, embora seja importante para os autores dos *e-books*, não será considerada para o modelo da linguagem de autoria. Isso porque a monetização pode ser realizada através de outras formas como na escolha da forma e plataforma de distribuição do material. A subcategoria *internet* (compartilhamento em redes sociais) foge ao escopo desta dissertação e também não será considerada no modelo da linguagem de autoria.

Tabela 1 – Estrutura final proposta pelos participantes da pesquisa sobre os elementos dos e-books infantis interativos

Ação	Gráficos	Áudio	Configuração	Navegação	Monetização	Jogos
pulos	zoom	player de som (iniciar, pausar, avançar, retroceder, aceitar)	menu de cores	paginação (folheamento, setas)	propaganda	colorir
gestos (transições)	texto	comandos de voz	idiomas	internet (on/off, compartilhamento em redes sociais)	vendas de histórias	memória
toque	personagens	volume (aumentar/diminuir, mudo, silenciar falas, músicas e efeitos)	cenas	marca páginas		quebra cabeça
acelerômetro (sacudir, balançar elementos)	cenários	músicas	falas	seleção de histórias		caça palavras
explosões	imagens (desenhos, figuras, fotos, ilustrações)	efeitos	iluminação (contraste, brilho)	menu inicial		
vídeos	efeitos intuitivos	narrações (descritivas, textos, falas)	fontes (tipos, tamanhos)			
movimentos			status do sistema			
			acessibilidade (líbras, áudio descrição)			

Fonte: adaptado de “Padrões de Interação para Ferramentas de Autoria de Storytellings infantis” (CRUZ, 2016)

As categorias Ação, Gráficos e Áudio, serão revisadas pois estão muito abrangentes, misturando interações do usuário (gestos e toque) e mídias (imagens, vídeos, áudios e "explosões"). Por exemplo, cenários e personagens classificados como entidades de primeira ordem nessa estrutura, possivelmente por uma visão dos participantes mais voltada ao roteiro da história, na verdade podem ser subtipos de imagens que já inclui desenhos, figuras, fotos e ilustrações. O zoom, explosões e "efeitos intuitivos" que não ficam totalmente claros, mas possivelmente se enquadrariam melhor na categoria ação como efeitos em geral (efeitos de imagem).

A categoria Configuração passa a envolver as configurações particulares ao *e-book* ou configurações globais do dispositivo como o controle de volume do áudio, menus de idiomas, e iluminação do sistema. Assim, alguns itens dessa categoria serão removidos tais como a subcategoria falas, que poderia ser englobada como um tipo especial nas subcategorias áudio, texto e imagens dependendo do tipo de falas a que se refere (narração ou legendas).

Esses problemas podem ser justificados pela forma que a estrutura foi concebida. Isso porque, no grupo de foco, os usuários possivelmente relataram suas percepções de uso do aplicativo e não a forma como eles criariam esses *e-books*.

2.2 Catálogo de livros

Esta seção apresenta uma análise crítica de alguns livros interativos com o objetivo de identificar e classificar as principais características de livros infantis interativos. A escolha dos livros a seguir foi realizada por meio de buscas por aplicativos para *smartphones* e *tablets* usando as palavras chaves: *histórias interativas*, *contos infantis*, *histórias para crianças* e *storytelling* interativo. Os resultados mais relevantes ou que apresentavam características únicas são apresentados a seguir. Os *e-books* infantis interativos apresentam contos e histórias voltados para o público infanto-juvenil. Assim, para a seleção focou-se aplicativos multimídia interativos, que apresentam um conto, fábula, ou história, usando recursos interativos que enriqueçam a experiência da leitura, tornando-a mais dinâmica e atraente a esse público.

As características avaliadas foram o tipo de narrativa, mídias, manipulação de mídias, controles da narrativa, formas de interação, e *widgets*. Essas categorias foram pensadas adaptando e reorganizando os elementos apontados na Seção 2.1.

- Tipos de narrativas: refere-se ao forma do roteiro da narrativa, se é linear ou não-linear;
- Mídias: são quais tipos de mídia são usados no *e-book*;

- Manipulação de mídias: atributos das mídias que podem ser manipulados, tanto pelo leitor quanto programaticamente pelo autor do *e-book*;
- Controle da narrativas: controles disponíveis para a navegação, controle do fluxo da narrativa, controle de configuração da apresentação ou do dispositivo;
- Formas de interação: recursos disponíveis para a interação com usuários, comandos de voz, sensibilidade ao toque, sensores de movimento, etc;
- *Widgets*: sub sistemas complementares à história, tais como mini-jogos.

Nas subseções seguintes são descritos exemplos de *e-books* interativos encontrados para Android, IOS e Web.

2.2.1 *Elements of a Story* - Cinderella

*Elements of a Story*³ é um web site interativo onde estudantes podem aprender sobre os diferentes "ingredientes" que compõem uma história, usando o conto *Cinderella* como exemplo. As atividades do site são voltadas para crianças que, após experimentar uma versão interativa da história, são apresentados vários conceitos comuns para histórias: cenário, personagens, enredo, exposição, conflito, clímax e resolução.

O site é uma produção da *Thirteen/WNET New York*, com a distribuição da *Annenberg Learner*. A *Annenberg Learner* se propõe como missão melhorar a qualidade de ensino nas escolas americanas, financiando e distribuindo recursos multimídia para professores. O site visa auxiliar professores a verificar quais as dificuldades dos alunos com a compreensão dos conceitos fundamentais das histórias.

A Figura 2 apresenta o *layout* do site. O menu principal do site apresenta atalhos para a introdução da história, a apresentação da história em si, e por fim um teste de aprendizado para o aluno. A apresentação da história é dividida em oito partes que podem ser acessadas pelo menu secundário do site.

A primeira parte (*Cinderella*) apresenta uma versão interativa do conto de fadas, com texto, ilustrações e animações. Ainda há a narração da história que pode ser silenciada pelo usuário. A história apresenta quatro botões de controle: voltar (*back*), pausar/continuar (*pause/play*), próximo (*next*), mudo (*mute*). A história prossegue automaticamente a menos que seja pausada. O site alerta que se o botão mudo for acionado, a narração não é pausada, ela prossegue normalmente só que no modo silencioso.

As outras sete partes do site focam em conceitos individuais dos elementos das histórias. Em cada parte é apresentada uma breve explicação de cada elemento seguindo-se

³ <https://www.learner.org/interactives/story/cinderella.html>

Figura 2 – Layout do site *Elements of a Story*

Fonte: screen shot da página *Elements of a Story*.

de uma atividade em formato de pequenos jogos, que solicitam ao estudante escolher os personagens, ou as repostas corretas de acordo com o conceito apresentado.

A Figura 3 mostra dois exemplos de jogos presentes no site *Elements of a Story*. O primeiro jogo solicita que o estudante escolha entre as figuras apresentadas, os personagens da história *Cinderella*. No segundo jogo é necessário que arraste os personagens e características para as áreas corretas especificadas na tela. Esses dois jogos representam as principais mecânicas dos demais jogos do site.

As características encontradas nesse *e-book* são resumidas na Tabela 2.

Tabela 2 – Características presentes no *e-book Elements of a Story - Cinderella*

Características	<i>Elements of a Story - Cinderella</i>
Tipo de narrativa	linear
Mídias	imagens, animações, áudio, texto, partículas
Manipulação de mídias	posição, rotação, opacidade/transparência, tamanho
Controle de narrativas	navegação (próximo, anterior), pausa/parar, play, silenciar, autoplay, menu inicial
Formas de interação	mouse (clique)
<i>widgets</i>	jogos (ligar pontos, selecionar resposta, memória)

Fonte: elaborado pelo autor.

2.2.2 *Tales with gigi*

Tales with Gigi é um aplicativo de *storytelling* que fornece uma biblioteca de contos de fadas para crianças tais como Branca de neve, Cinderella, A Pequena Sereia, entre outras. Esse aplicativo pode ser encontrado na *PlayStore* (loja de aplicativos para do sistema Android).

Figura 3 – Dois exemplos de jogos dos exercícios de fixação encontrados no site *Elements of a Story*

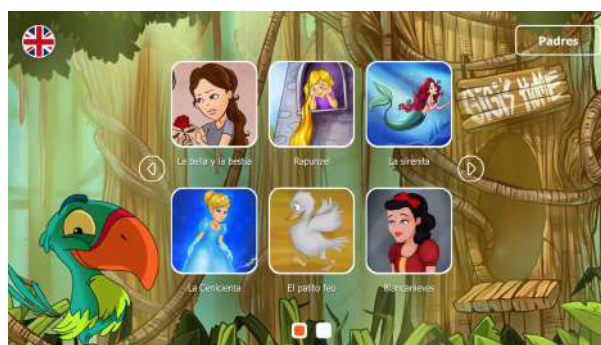


Fonte: *screen shot* da página *Elements of a Story*.

O desenvolvedor ⁴ do aplicativo também disponibiliza os contos em aplicativos separados em sua página na loja. A descrição do aplicativo afirma que já houve mais de duzentos mil downloads mundialmente, e que a interface foi planejada especialmente para crianças.

A Figura 4 mostra o menu inicial do aplicativo. Esse menu apresenta com destaque a seleção das histórias. É possível também selecionar um idioma alternativo para o aplicativo usando o botão no canto superior esquerdo.

Figura 4 – Menu inicial do aplicativo *Tales with Gigi*

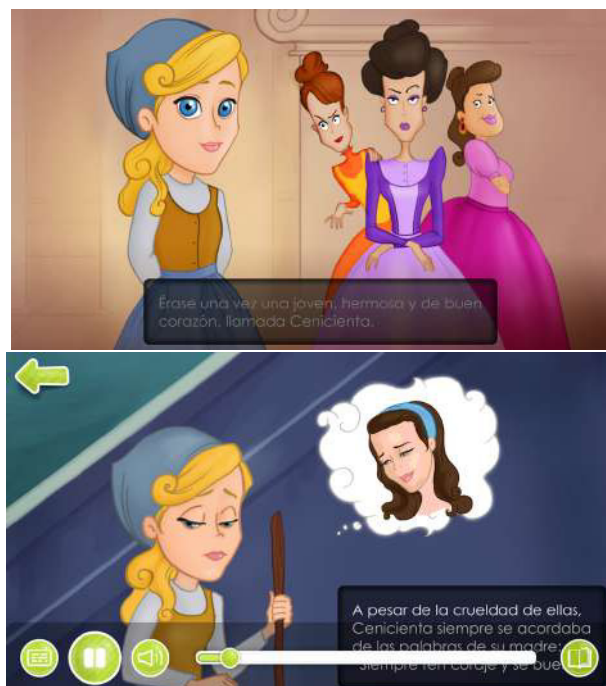


Fonte: *screen shot* do aplicativo.

⁴ A página do desenvolvedor do aplicativo *Tales with Gigi* pode ser acessada pelo link <https://play.google.com/store/apps/dev?id=6290881108498971005>

As histórias são apresentadas em tela cheia e, ao serem tocadas, apresentam os botões de controle e uma linha de tempo como mostrado na Figura 5 onde se vê as telas com a apresentação normal e com os controles ativados pela interação do usuário.

Figura 5 – História da *Cinderella*, com os botões de controle ocultos e com os controles apresentados



Fonte: *screen shot* do aplicativo.

No canto superior esquerdo podemos ver o botão para voltar ao menu inicial do aplicativo. Na região inferior da tela estão os controles de navegação. Da esquerda para a direita temos os seguintes botões: *mostrar/ocultar legendas*, *pausar/continuar*, *volume da narração*, *linha do tempo*, e *reprodução automática*.

Os livros interativos nesse aplicativo apresentam como elementos interativos somente os botões do controle de navegação. Existe a narração completa da história e o texto é usado como um elemento secundário para acompanhar a narração como se fosse a legenda. No quesito de elementos gráficos existem algumas animações simples, mas são usados como detalhe. O destaque da navegação dessas histórias está na linha do tempo que permite ao usuário "saltar" para qualquer momento da história.

Ao término da história, o usuário pode jogar um dos dois jogos correspondentes à história. A Figura 6 mostra os jogos disponíveis no aplicativo que correspondem à história da *Cinderella*. O primeiro jogo é um jogo de memória que o usuário tem que combinar as quatro partes da figura correta. O segundo jogo corresponde a um jogo de quebra cabeças.

As características encontradas nesse *e-book* são resumidas na Tabela 3.

Figura 6 – Jogos disponíveis no aplicativo *Tales with Gigi*

Fonte: *screen shot* do aplicativo.

Tabela 3 – Características presentes no e-book *Tales with gigi*

Características	<i>Tales with gigi</i>
Tipo de narrativa	linear
Mídias	imagens, áudio, texto, animação
Manipulação de mídias	posição, rotação, opacidade/transparência, tamanho, volume do áudio
Controle de narrativas	navegação (próximo, anterior), pausa/parar, play, silenciar, autoplay, menu inicial, Seleção de conto, seleção de idioma
Formas de interação	toque
<i>widgets</i>	jogos (Quebra-cabeças, colorir)

Fonte: elaborado pelo autor.

2.2.3 *Classic fairy tales*

O aplicativo *Classic fairy tales* conta com uma biblioteca com vários contos infantis. Como visto na Figura 7 ele disponibiliza essas histórias traduzidas nas seguintes línguas: inglês, espanhol, chinês, francês, alemão, português, russo ou valenciano.

Esse aplicativo pode ser encontrado na *PlayStore*⁵ na categoria entretenimento. O desenvolvedor desse aplicativo possui em sua página na loja outros vários aplicativos de contos infantis, inclusive dois outros aplicativos intitulados *Classic Fairy Tales 2*⁶ e

⁵ <https://play.google.com/store/apps/details?id=es.cuentuvere.cuentosclasicos>

⁶ <https://play.google.com/store/apps/details?id=es.cuentuvere.cuentosclasicosdos>

Figura 7 – Menu inicial do aplicativo *Classic Fairy Tales* para seleção do idioma

Fonte: *screen shot* do aplicativo.

*Classic Fairy Tales 3*⁷ que apresentam comportamento semelhantes mas contêm coletâneas de histórias diferentes.

A Figura 8 mostra a tela da história *Cinderella*. Todas as telas da história apresentam o mesmo leiaute: com o texto e botões de controle à esquerda e uma ilustração à direita.

Figura 8 – Tela da história *Cinderella* no aplicativo *Classic Fairy Tales*

Fonte: *screen shot* do aplicativo.

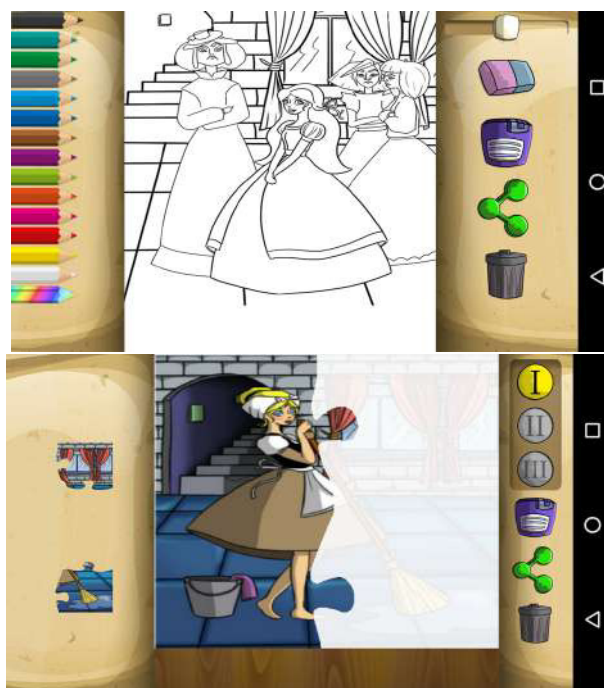
Esse aplicativo apresenta um botão que permite que o usuário aumente o tamanho da fonte durante sua apresentação. Isso ajuda a leitura para pessoas com dificuldades visuais ou em dispositivos muito pequenos.

Há dois jogos que podem ser acessados a qualquer momento pelo usuário usando os botões logo abaixo do texto da história. A Figura 9 mostra exemplos desses dois jogos.

O primeiro jogo é o para o usuário colorir uma ilustração como desejar, ou usar o "lápiz arco-iris" para pintar com as cores sugeridas pelo aplicativo. Já o segundo jogo apresenta um quebra cabeças com três níveis de dificuldades.

As características encontradas nesse *e-book* são resumidas na Tabela 4.

⁷ <https://play.google.com/store/apps/details?id=es.cuentuvere.cuentosclasicostres>

Figura 9 – Exemplo de jogos no aplicativo *Classic Fairy Tales*

Fonte: *screen shot* do aplicativo.

Tabela 4 – Características presentes no *e-book Classic fairy tales*

Características	<i>Classic fairy tales</i>
Tipo de narrativa	linear
Mídias	imagens, áudio, texto
Manipulação de mídias	volume do áudio
Controle de narrativas	pausa/parar, play, silenciar, menu inicial, Seleção de conto, seleção de idioma
Formas de interação	toque
<i>widgets</i>	jogos (Quebra-cabeças, colorir)

Fonte: elaborado pelo autor.

2.2.4 Livro mágico

Esse aplicativo conta com uma biblioteca de dois títulos: O galinho Zé e O menino da capa vermelha. A Figura 10 mostra a tela de seleção das histórias desse aplicativo.

O aplicativo está todo em português. Ele não possui narração, efeitos sonoros, nem animações, deixando a desejar nas opções de interação com o usuário para tornar a leitura mais cativante. Ele possui um roteiro não linear onde o usuário pode escolher diferente caminho alterando sua experiência na história, como apresentado na Figura 11.

A navegação do aplicativo também não permite que o usuário volte para uma cena anterior somente podendo então avançar ou voltar ao menu inicial.

As características encontradas nesse *e-book* são resumidas na Tabela 5.

Figura 10 – menu de seleção dos contos no aplicativo Livro mágico



Fonte: *screen shot* do aplicativo.

Figura 11 – Um dos desvios onde o usuário pode decidir como a história deve prosseguir



Fonte: *screen shot* do aplicativo.

Tabela 5 – Características presentes no *e-book* Livro mágico

Características	Livro mágico
Tipo de narrativa	não-linear
Mídias	imagens, texto
Manipulação de mídias	-
Controle de narrativas	navegação (próximo), menu inicial, Seleção de conto
Formas de interação	toque
<i>widgets</i>	-

Fonte: elaborado pelo autor.

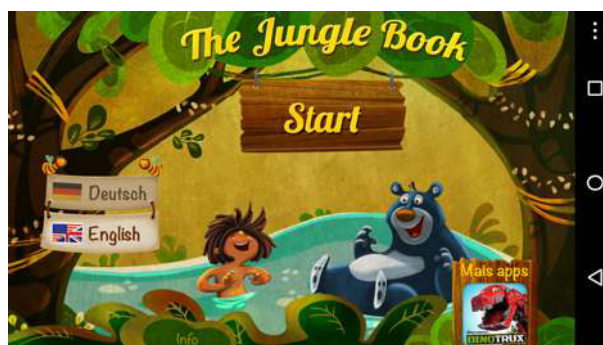
2.2.5 *The Jungle Book*

Esse aplicativo conta a história de Mogli. O livro “*The Jungle Book*” do autor Rudyard Kipling é uma história infantil clássica. O desenvolvedor desse aplicativo cita que reinterpretou essa história clássica para adaptá-la a um contexto mais moderno no formato de um livro interativo.

A história está disponível em duas línguas, alemão e inglês, que pode ser escolhida ao iniciar no menu inicial. A Figura 12 apresenta o menu inicial desse aplicativo.

O aplicativo não apresenta texto, e é completamente narrado. A navegação entre cenas se dá de forma automática, mas o usuário pode escolher passar para próxima tela

Figura 12 – Um dos desvios onde o usuário pode decidir como a história deve prosseguir



Fonte: *screen shot* do aplicativo.

com o gesto "*swipe*". Ele apresenta várias animações e elementos interativos em cada cena, onde o usuário pode tocar e interagir com os personagens. Nas cenas tem um menu na parte inferior da tela que permite silenciar a narração, reiniciar a narração da cena atual ou voltar para o menu inicial.

As características encontradas nesse *e-book* são resumidas na Tabela 6.

Tabela 6 – Características presentes no *e-book The Jungle Book*

Características	<i>The Jungle Book</i>
Tipo de narrativa	linear
Mídias	imagens, áudio, texto
Manipulação de mídias	posição, rotação, opacidade/transparência, tamanho, volume do áudio
Controle de narrativas	navegação (próximo, anterior), pausa/parar, play, silenciar, autoplay, menu inicial, seleção de idioma
Formas de interação	toque
<i>widgets</i>	-

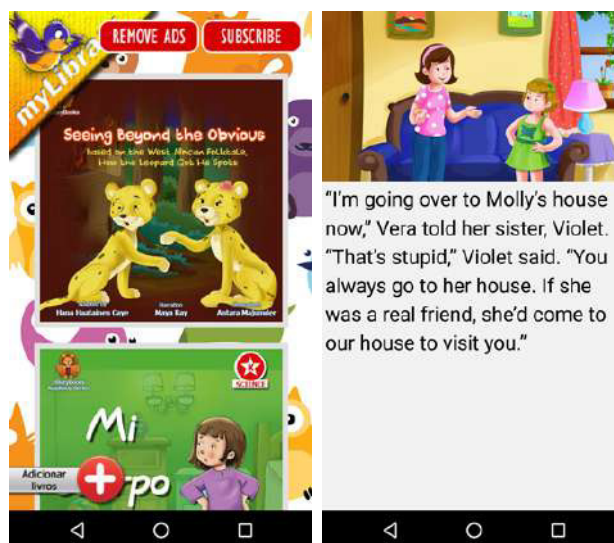
Fonte: elaborado pelo autor.

2.2.6 *iStoryBooks*

O aplicativo *iStoryBooks*⁸ conta com uma biblioteca com vários contos infantis. A Figura 13 mostra dois *screen shots* do aplicativo a tela de seleção das histórias e uma página da história “Seeing beyond the obvious”. Os contos apresentados no aplicativo são simples, cada página contém uma ilustração o texto e a narração da história sincronizada às páginas. O aplicativo possui também um menu para o usuário ativar/desativar a narração e passagem automática das páginas.

As características encontradas nesse *e-book* são resumidas na Tabela 7.

⁸ <https://play.google.com/store/apps/details?id=com.infomarvel.istorybooks>

Figura 13 – Aplicativo *iStoryBooks*

Fonte: *screen shot* do aplicativo.

Tabela 7 – Características presentes no e-book *iStoryBooks*

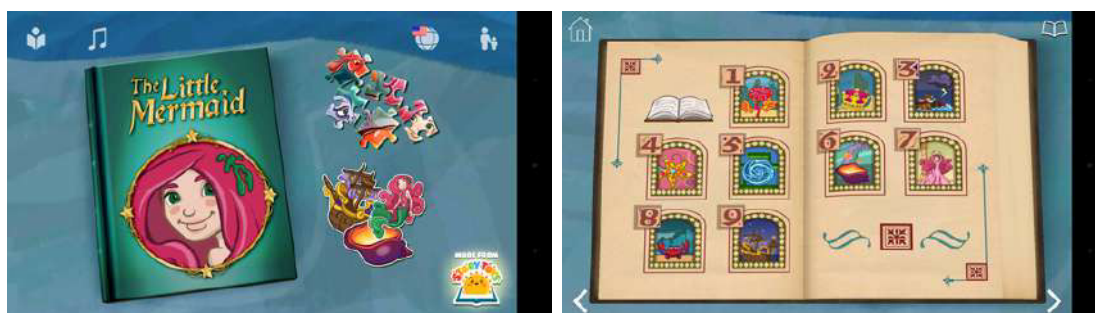
Características	<i>iStoryBooks</i>
Tipo de narrativa	linear
Mídias	imagens, áudio, texto
Manipulação de mídias	-
Controle de narrativas	navegação (próximo, anterior), silenciar, reprodução automática, menu inicial, múltiplas histórias
Formas de interação	toque
<i>widgets</i>	-

Fonte: elaborado pelo autor.

2.2.7 *The Little Mermaid*

O aplicativo *The Little Mermaid* ⁹ apresenta a história da pequena sereia. A Figura 14 mostra os principais menus do aplicativo: o menu inicial, e o menu de seleção de capítulos. O aplicativo possui traduções para os idiomas inglês, francês, alemão, espanhol e dinamarquês. No menu principal é possível escolher entre começar a história ou duas atividades extras: um jogo de quebra-cabeças ou montar um cenário com “sticks”.

As características encontradas nesse e-book são resumidas na Tabela 8.

Figura 14 – e-book *The Little Mermaid*

Fonte: *screen shot* do aplicativo.

Tabela 8 – Características presentes no e-book *The Little Mermaid*

Características	<i>The little Mermaid</i>
Tipo de narrativa	linear
Mídias	imagens, áudio, texto
Manipulação de mídias	posição, rotação, opacidade/transparência, tamanho
Controle de narrativas	navegação (próximo, anterior), silenciar, reprodução automática, menu inicial, seleção de idioma, seleção de capítulo
Formas de interação	toque
<i>widgets</i>	jogos

Fonte: elaborado pelo autor.

2.3 Características de e-books interativos

2.4 Considerações finais

Neste capítulo foram apresentados os requisitos e padrões de interação de e-books interativos levantados pela pesquisa de Cruz (2016). Esses requisitos não se mostraram suficientemente prontos para ser aplicados na construção da linguagem e foi necessário adaptá-los. Essas adaptações aproximaram sua estrutura ao ambiente de desenvolvimento.

Assim, com base na Tabela 1 e análise de novos e-books interativos do catálogo apresentado, tem-se essa nova lista de características para e-books infantis interativos apresentada na Tabela 9.

⁹ <https://play.google.com/store/apps/details?id=com.storytoys.TheLittleMermaid.GooglePlay>

Tabela 9 – Requisitos revisados para e-books interativos

Características	<i>Cinderella</i>	<i>Tales with Gigi</i>	<i>Classic Fairy Tales</i>	Livro mágico	<i>The Jungle Book</i>	<i>iStory-Books</i>	<i>The Little Mermaid</i>	Total
Narrativas								
linear	x	x	x		x	x	x	85,7%
não-linear				x				14,3%
Mídias								
imagem	x	x	x	x	x	x	x	100%
áudio	x	x	x		x	x	x	85,7%
texto	x	x	x	x	x		x	85,7%
vídeo								0%
animação	x	x			x		x	57%
partículas	x							14,3%
Manipulação de mídias								
opacidade	x	x			x		x	57%
posição	x	x			x		x	57%
rotação	x	x			x		x	57%
tamanho	x	x			x		x	57%
Controles da narrativa								
botões de navegação (iniciar, parar, próximo, anterior etc)	x	x	x		x		x	71,4%
menu inicial	x	x	x	x	x	x	x	100%
controle de volume	x		x		x		x	57%
Parental Control							x	14,3%
múltiplas histórias		x	x	x	x	x		71,4%
seleção de idioma		x	x		x		x	57%
Formas de interação								
toque/gestos		x	x	x	x	x	x	85,7%
acelerômetro								0%
giroscópio								0%
microfone								0%
câmera								0%
mouse	x							14,3%
Widgets								
jogos	x	x	x				x	57%

Fonte: elaborado pelo autor.

3 Tecnologias Relacionadas

Neste capítulo são discutidos trabalhos que objetivam melhorar e/ou enriquecer os *e-books* interativos ou facilitar sua autoria. Estes trabalhos apresentam diferentes abordagens quanto à forma de apresentar essas melhorias. Destas abordagens pode-se destacar quatro: propor uma extensão em uma linguagem de autoria de *e-book* já consolidada para adicionar ou alterar suas características e/ou funcionalidades; propor uma aplicação em código nativo como prova de conceito para explorar novas funcionalidades para *e-books* interativos; propor uma ferramenta de autoria para auxiliar os autores na criação de elementos mais complexos; propor uma nova linguagem de autoria para criação de *e-books* interativos.

Extensões em linguagens permitem adicionar ou remover funcionalidades. Quando feito da forma correta isso permite fazer uso da popularidade de uma linguagem e seu “ecossistema” e ferramentas já estabelecido tornando-a mais adequada a sua necessidade. Muitas vezes as extensões em uma linguagem são pequenas e focam em carências que a linguagem apresenta. Nestes casos, geralmente, essas extensões são incorporadas à linguagem como padrão em uma versão futura, como pode ser visto pela inclusão de componentes multimídia nativos em HTML5 (HICKSON et al., 2014).

Entretanto estender uma linguagem geralmente envolve um rompimento com a linguagem original. Isso muitas vezes exige um trabalho maior criando ferramentas, *plugins*, ou *players* compatíveis para dar suporte à versão estendida da linguagem. Além disso, quanto maior forem as mudanças à linguagem, maior é esse trabalho. Assim algumas vezes as linguagem estendidas se tornam incompatíveis com a linguagem original e suas ferramentas e *players* atualmente no mercado como é o caso do *iBooks* (BOTT, 2012) (GLAZMAN, 2012) (BJARNASON, 2012).

Contudo a Apple¹ com a nova versão da ferramenta de autoria *iBooks Author*² (ALBANESIUS, 2012) propõe uma extensão para o formato EPUB 3. Esse novo formato que foi criado focando no sistema IOS e acrescenta novas extensões CSS (W3C, 2001) que não fazem parte da recomendação da W3C e usa um XML *namespace* proprietário (BJARNASON, 2012) (GLAZMAN, 2012). As principais funcionalidades adicionadas por esse novo modelo são a criação de leiautes fixos para as páginas, *widgets*, e um controle maior do estilo de apresentação do texto.

Por sua vez, aplicativos nativos são uma forma prática de apresentar provas de conceito para *e-books* interativos. O tempo necessário para desenvolver uma aplicação é relativamente curto se comparado com o exigido para criar uma nova ferramenta de

¹ <http://www.apple.com/>

² <http://www.apple.com/br/ibooks-author/>

autoria, garantir a compatibilidade de extensões com uma linguagem ou criar o formalismo para uma nova linguagem.

O problema com essa abordagem é a dificuldade de criar novos *e-books*. Por exemplo, mesmo que o uso de linguagens de programação nativas das plataformas alvo permitam mais liberdade quanto aos recursos, a forma do *e-book* interativo elas não fornecem ferramentas e recursos para simplificar a criação em série de diferentes aplicativos desse tipo. Neste caso o apoio de uma linguagem apropriada para esse tipo de produto e suas ferramentas simplificam o processo de criação de múltiplos *e-books*.

Mori et al. (2011), por exemplo, desenvolveu um aplicativo para dispositivos móveis para apresentar um novo conceito denominado de *Interactive Rich Reading*. Esse aplicativo é um *e-book* interativo caracterizado por um agente conversacional baseado em vídeo que faz perguntas e comentários sobre a página da história atual para motivar e enriquecer a experiência de leitura infantil. O principal desafio para emular esse tipo de *e-book* em uma das linguagens de autoria padrão atuais é a sobreposição do vídeo com transparência. O autor relata que embora o padrão de codificação de vídeo H.264 (WIEGAND et al., 2003) oficialmente dê suporte à transparência, quase nenhuma implementação apresenta essa funcionalidade. Assim, para obter esse resultado ao sobrepor o vídeo com o *e-book*, foi usada uma técnica alternativa para codificar a transparência juntamente com o vídeo e depois resolver isso em tempo de processamento para cada quadro do vídeo.

As próximas seções exploram mais detalhadamente formas de criar e editar *e-books* interativos. Na Seção 3.1 são apresentadas ferramentas para autoria de *e-books* ou com propósito semelhante. Na Seção 3.2 são apresentadas linguagens de autoria multimídia. Os trabalhos discutidos neste capítulo servem de inspiração para a proposta do modelo conceitual apresentado no Capítulo 4.

3.1 Ferramentas de autoria

Como vantagem de criar uma ferramenta de autoria pode-se citar a facilidade de aplicar os conceitos usados para sua interface de usuário a diferentes linguagens ou formatos semelhantes. Mesmo quando as ferramentas são planejadas com uma linguagem ou formato específico em mente, elas podem ser reutilizadas pra outros formatos semelhantes com pouca ou nenhuma dificuldade. Isso pode ser percebido pela facilidade em que as ferramentas trabalham com mais de uma linguagem ou pelo menos exportam produto final em diferentes formatos.

Contudo, ferramentas de autoria apresentam a desvantagem de sempre serem limitadas pelas linguagens e formatos que desejam editar. Isso se dá pelo fato de que ao terminar a edição, o produto final gerado tem que seguir o padrão definido pela linguagem ou formato para ser compatível e executável pelos *players* daquela linguagem.

Assim ferramentas de autoria podem simplificar o trabalho dos autores na criação de comportamentos ou componentes complexos, mas esses recursos sempre estão limitados pelo que é permitido pela linguagem alvo.

Por exemplo, o artigo (SIGARCHIAN et al., 2013) propõe um novo ambiente de autoria para *e-books* interativos com o padrão EPUB 3 (CONBOY et al., 2011). O objetivo dessa proposta é explorar formas de reduzir o custo para as editoras da criação de *e-books* interativos. De acordo com a proposta, tal ambiente de autoria tem que possuir uma interface de usuário que possa ser operada por não programadores. Outro foco da proposta é de criar *e-books* enriquecidos com interatividade e multimídia totalmente compatíveis com o padrão EPUB 3. Assim, a solução proposta é que o ambiente de autoria permita a criação, armazenamento e até mesmo compartilhamento desses “enriquecimentos” em forma de *widgets* para reuso posterior em outros *e-books*.

Nas subseções seguintes são apresentadas treze ferramentas de autoria detalhando seus objetivos, público-alvo e funcionalidades.

3.1.1 *iBooks Author*

*iBooks Author*³ é uma ferramenta de autoria de livros digitais. Ela permite criar e exportar livros nos formatos PDF (INCORPORATED, 2006a), EPUB (CONBOY et al., 2011), *iBooks*, arquivo de texto (sem formatação) ou também é possível publica-los para a loja de livros da *iBooks*. O formato *iBooks* é uma variação proprietária do padrão EPUB (CONBOY et al., 2011), que foi adaptado para dispositivos que rodam o sistema IOS.

Essa ferramenta é *What You See is What You Get* (WYSIWYG) termo usado para classificar ferramentas que permitem editar o “produto” no formato semelhante ao que será obtido como produto final. Para iniciar a criação de um livro é necessário escolher um dos modelos disponibilizados na ferramenta ou começar por um modelo em branco. Os modelos são divididos em três grupos relativos a seu leiaute: vertical, horizontal ou EPUB (leiaute adaptável). Os leiaute vertical e horizontal tem navegação fixa, ou seja, com páginas de tamanhos definidos e o leiaute EPUB tem navegação por rolagem. A escolha do leiaute também influencia no formato de arquivo que pode ser exportado. Os leiautes fixos podem ser exportados para os formatos *iBook* e PDF, enquanto o leiaute EPUB somente pode ser exportado para o formato EPUB.

A ferramenta também permite acrescentar formas (desenhos geométricos), gráficos, tabelas, caixas de texto, *widgets* e mídias (áudio, imagem, vídeo) em qualquer área do livro. A ferramenta disponibiliza alguns *widgets* inicialmente, como galeria de fotos, animações, *Pop-Over*, objetos 3D, etc. Caso deseje, o autor pode criar e distribuir seu próprio *widget*, que consiste de um pacote (*bundle*) com a extensão *wdgt*⁴. O arquivo principal do *widget*

³ <http://www.apple.com/ibooks-author/>

⁴ <<https://support.apple.com/en-us/HT204433>>

deve ser um arquivo HTML (extensão .html) combinado com Javascript e CSS. O HTML estrutura o widget enquanto o CSS estiliza e o Javascript acrescenta interatividade.

Os livros criados com o *iBooks Author* utilizam a tecnologia *VoiceOver* que auxilia pessoas com deficiência visual. Para usar a tecnologia *VoiceOver* com widgets e mídias é necessário, somente, acrescentar descrições de acessibilidade.

3.1.2 BlueGriffon EPUB Edition

BlueGriffon ⁵ é uma ferramenta WYSIWYG para edição de páginas web que possui uma versão para criação de *e-books* interativos em EPUB. Ela pode ser executada nos sistemas Windows, Mac, Linux e segue os padrões da *World Wide Web Consortium* (W3C). Ela pode criar e editar páginas em HTML 4, XHTML 1.1, HTML 5 e XHTML 5, também suporta CSS 2.1 e CSS 3.

Ela possui três tipos de licença: sem licença (*free*), licença básica, e licença EPUB. A licença EPUB, além das funcionalidades das outras licenças, pode criar e editar *e-books* nos formatos EPUB 2, EPUB 3.0.1, EPUB 3.1, e suporte a edição dos metadados.

A ferramenta dá suporte à edição básica de EPUB mas, para implementar funcionalidades mais complexas, é necessário codificar em JavaScript ([ECMAScript Association et al., 2016](#)), CSS ([W3C, 2001](#)) e HTML ([Hickson et al., 2014](#)).

3.1.3 CircularFLO

CircularFLO ⁶ é uma ferramenta para criar *e-books* interativos com leiaute fixo em EPUB 3 ou KF8. A ferramenta converte arquivos InDesign ⁷ ou PDF e cria cópias fiéis em formatos padrões de *e-books* textual.

A ferramenta permite criar *e-books* com narração, sincronizando áudio ao texto, manualmente ou automaticamente através da edição pelo InDesign. Não tem restrições para o tamanho ou proporções das páginas, imagens usadas, fontes, ou espaço de cores (RGB ou CMYK). As animações são criadas usando CSS e podem ser adicionadas para imagens e textos. A ferramenta disponibiliza algumas animações em CSS que podem ser ajustadas, tanto o tempo as repetições e a duração. Para funcionalidades mais complexas é possível integrar códigos web nas páginas, bem como áudio e vídeo. Os vídeos podem ser incluídos *offline* no arquivo ou como *stream*.

⁵ <<http://bluegriffon.org/>>

⁶ <<https://www.circularflo.com/>>

⁷ <<https://www.adobe.com/br/products/indesign.html>>

3.1.4 TaleSpring

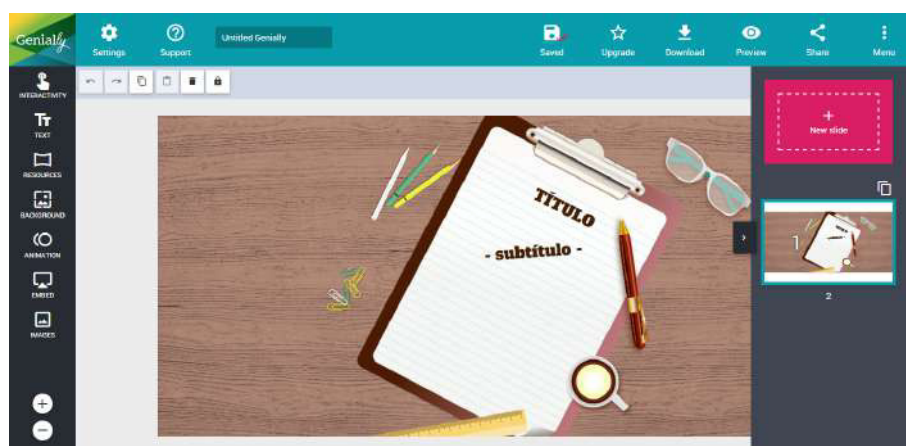
TaleSpring ⁸ é um sistema online para criação de *e-books* interativos. Ele permite criar histórias com animações, áudio, narração, e *highlight* de texto. A ferramenta é gratuita para uso, mas cobra por publicação e participação nos lucros (*royalties*).

3.1.5 Genially

Genially ⁹ é uma ferramenta de *storytelling* online para criar e editar imagens, infográficos, apresentações, *hotsites* etc. A ferramenta permite criar infográficos interativos e animados, e embutir vídeo, áudio, gráficos, mapas etc nas apresentações. Ela oferece uma interface WYSIWYG para edição. Após a edição, ela pode publicar a apresentação online pelo sistema ou fazer o download da mesma.

A ferramenta usa a abstração de slides para edição das histórias. A Figura 15 mostra a interface do sistema, onde no lado esquerdo tem o menu principal com os elementos que podem ser incluídos na apresentação e no lado direito tem-se o menu de navegação dos slides.

Figura 15 – Interface do sistema *Genially*



Fonte: *screen shot* do aplicativo.

3.1.6 My Story Book Creator School Edition

My Story Book Creator School Edition ¹⁰ é um aplicativo para iPhone e iPad que permite criar apresentações interativas voltado para crianças. Ele permite criar e compartilhar *e-books* e filmes através de *storytelling* adicionando desenhos, fotos etc. Ela apresenta a funcionalidade de gravação de voz que pode ser adicionada para as páginas. O aplicativo permite publicar as histórias na web, Facebook e Twitter, além de exportar

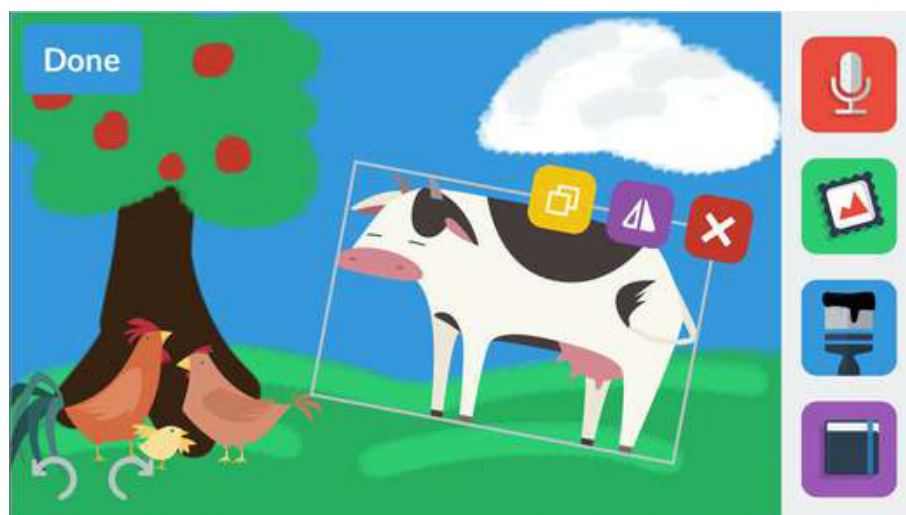
⁸ <<http://www.talespring.com/>>

⁹ <<https://www.genial.ly/pt-br>>

¹⁰ <<https://itunes.apple.com/us/app/my-story-book-maker-for-kids/id449232368>>

para o formato EPUB, PDF ou em vídeo MP4 diretamente no Youtube. A ferramenta foi modelada para ser simples e fácil de usar por crianças. A Figura 16 mostra um *screen shot* da interface do aplicativo.

Figura 16 – Interface do aplicativo *My Story Book Creator School Edition*



Fonte: *screen shot* do aplicativo retirado do site

<<https://itunes.apple.com/us/app/my-story-book-maker-for-kids/id449232368>>.

3.1.7 Kitaboo

Kitaboo ¹¹ é uma plataforma online para criação, publicação e distribuição de *e-books* interativos voltados para dispositivos móveis. Os *e-books* interativos criados por essa plataforma podem conter áudio, vídeos, animações e interação. O sistema usado para publicação pelo sistema permite gerenciar e coordenar autoria com múltiplos usuários inclusive autoria colaborativa em tempo real.

Kitaboo foi criado com o objetivo de criar conteúdo educativo, mesmo assim é possível criar o nível de interatividade necessário para *e-books* interativos infantis com as funcionalidades disponíveis. A plataforma também permite exportar para o formato EPUB.

3.1.8 Kwik

Kwik ¹² é um plugin para o Photoshop ¹³ Creative Suite 5 (CS5) ou Creative Suite 6 (CS6). Ele permite que designers gráficos e ilustradores criem aplicações interativas para dispositivos móveis diretamente do *Photoshop*.

¹¹ <<https://kitaboo.com/>>

¹² <<http://kwiksher.com/kwik/>>

¹³ <<https://www.adobe.com/br/products/photoshop.html>>

Para exportar ou visualizar o projeto funcionando é usado o Corona SDK ¹⁴. Após a edição, o projeto *Kwik* pode ser exportado na linguagem Lua na qual pode ser melhorado ou diretamente exportado para uma das lojas virtuais para aplicativos de dispositivos móveis.

Essa ferramenta permite criar *e-books* interativos com áudio, texto, imagens, além de ter narração, *highlight* do texto sincronizado, suporte a múltiplas linguagens para o texto e áudio, animações. Também é possível aprimorar o aplicativo com física simples para os objetos ou configurar funcionalidades como monetização.

3.1.9 Demibooks

Demibooks ¹⁵ é um aplicativo para IOS ¹⁶ para criar *e-books* interativos sem precisar de conhecimento prévio em programação. Ele permite importar as mídias diretamente no *tablet* ou *smartphone*, adicionar as animações e interações e organizar os conteúdos em páginas.

O aplicativo oferece como funcionalidades para os *e-books* os recursos de física, animações, e outros comportamentos mais complexos para as imagens, textos, sons e vídeos. Ele também fornece um serviço de publicação dos *e-books*. Os criadores do aplicativo citam como funcionalidades:

- Avançar e retroceder animações quadro-a-quadro;
- Física avançada, incluindo juntas, molas etc;
- Capacidade de rastrear e gerenciar objetos usando até 15 *hotspots*;
- Loops de comportamento simples;
- Efeitos de partículas;
- Opções de layout Retrato ou Paisagem;
- Configurar comportamentos complexos no Editor de Efeitos através de séries de instruções If-Then;
- Construa e organize rapidamente páginas, navegação e índice;
- Adicionar links web;
- Aplicar movimento, funções e espera (*delay*) à mídia;

¹⁴ <<http://www.coronalabs.com/products/corona-sdk/>>

¹⁵ <<http://demibooks.com/>>

¹⁶ <<https://itunes.apple.com/us/app/composer-pro/id537026880>>

- Tornar os objetos arrastáveis ou fazê-los reagir ao acelerômetro através de Física;
- Visualizar diretamente no iPad;
- Importar/exportar livros via iTunes;
- Compartilhar ou publicar seu aplicativo concluído.

3.1.10 Kindle Textbook Creator

Kindle Textbook Creator ¹⁷ é uma ferramenta que permite criar *e-books* para Kindle a partir de PDF existentes. O processo consiste do usuário abrir o arquivo PDF que ele deseja editar e adicionar imagens, áudio e vídeos. O resultado pode ser exportado para o formato padrão para Kindle, *Kindle books*. O aplicativo permite também publicar o *e-book* para o *Kindle Direct Publishing* (KDP) ¹⁸ e pode ser acessado em Fire tablets e aplicativos para iPads, Android tablets, smartphones, PCs, e Macs.

3.1.11 Kindle Comic Creator

Kindle Comic Creator ¹⁹ é uma ferramenta com o objetivo de incluir elementos interativos a revistas em quadrinho e torná-las um formato compatível com o *Kindle*. Ela detecta e recomenda automaticamente o posicionamento da Visualização do Painel do Kindle no seu livro. Como principais funcionalidades são citadas:

- Navegação guiada com *Kindle Panel View*;
- Livros com páginas duplas;
- Importar imagens nos formatos jpg, pdf, tiff, png e ppm;
- *Preview* do conteúdo em dispositivos Kindle antes de publicar.

3.1.12 Kindle Kids' Book Creator

Kindle Kids' Book Creator ²⁰ é mais uma ferramenta para criar *e-books* infantis para dispositivos Kindle. Essa ferramenta suporta vários layouts para livros infantis e até permite importar de um PDF e transformá-lo em um livro do Kindle.

¹⁷ <<https://www.amazon.com/gp/feature.html?docId=1002998671>>

¹⁸ <<https://kdp.amazon.com/>>

¹⁹ <<https://www.amazon.com/gp/feature.html?docId=1001103761>>

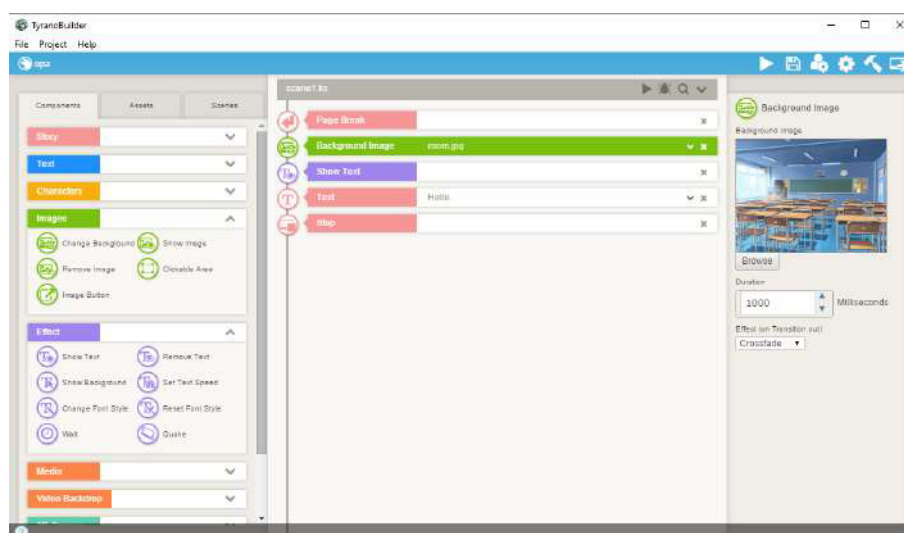
²⁰ <<https://www.amazon.com/gp/feature.html?docId=1002979921>>

3.1.13 TyranoBuilder

TyranoBuilder ²¹ é uma ferramenta para autoria de *visual novels*. A Figura 17 apresenta um *screen shot* do aplicativo TyranoBuilder. Na esquerda têm-se os menus de *Components*, *Assets* e de *Scenes*. O menu de *Components* apresenta os elementos que podem ser acrescentados na cena. O menu de *Assets* é onde os arquivos (imagens, áudio, etc) são importados. O menu *Scenes* tem a lista de cenas.

Na área central do aplicativo fica uma espécie de linha de tempo que determinar como será o comportamento da cena. Ela usa uma interface *drag-and-drop* para organizar os componentes nessa linha do tempo. Na área à direita são apresentados os detalhes do componente selecionado atualmente.

Figura 17 – Interface do aplicativo *TyranoBuilder*



Fonte: *screen shot* do aplicativo.

Ao terminar a edição, a visual novel pode ser exportada para aplicações web (*Browser game*), aplicativos nativos para Windows ou Mac, ou aplicativos móveis para Android ou IOS. A ferramenta permite importar mídias como imagens, áudio, vídeo, e editar texto. ela também permite ao autor usar *scripts* para funcionalidades mais complexas.

3.2 Linguagem de autoria

Como dito, algumas vezes extensões de linguagens populares acabam por se tornar linguagens de autoria independentes. Isso se mostra necessário quando se encontra uma nova forma de abordar a autoria, como um novo paradigma, de modo a entrar em conflito com a linguagem original. Ao criar novas linguagens, é possível explorar novas abstrações

²¹ <<http://tyranobuilder.com/>>

e oferecer novos recursos que não existem nas linguagens disponíveis no momento de sua criação.

Miller e Ranum (2012), seguem a abordagem de criar uma nova linguagem para autoria de *e-books* interativos. Esta linguagem tem por objetivo ajudar estudantes de ciência da computação no aprendizado de algoritmos. Os autores também incluíram um sistema de publicação código aberto baseado no Sphinx²² para incentivar a colaboração entre os autores. Nessa linguagem de autoria, os *e-books* são compostos por vídeos, texto e blocos de códigos. Os blocos de código podem então ser editados e executados pelo aluno diretamente no livro. Ainda há a ferramenta chamada *codlens* que é um visualizador de código que permite o aluno acompanhar a execução do código observando o resultado linha a linha. Esses *e-books* são então exportados em HTML (W3C, 2002) e Javascript (ECMAScript; ASSOCIATION et al., 2016).

Todavia a criação e popularização de uma nova linguagem de autoria envolve muito tempo e esforço. Além disso é necessário criar uma ambiente de suporte à autoria como *players*, ferramentas de autoria etc. O modelo conceitual Fábulas utiliza vários pontos dessas linguagens como inspiração para a sua estrutura. Nesta seção são apresentadas algumas dessas linguagens discutindo os seus pontos principais.

3.2.1 EPUB

O padrão EPUB (CONBOY et al., 2011) é um formato de código aberto para *e-books* que foi criado pelo *International Digital Publishing Forum* (IDPF)²³. A versão atual desse padrão é o EPUB 3.0 que agora é baseado em HTML5 (HICKSON et al., 2014) permitindo o uso de novos tipo de conteúdos multimídia nos *e-books* em comparação com o padrão antigo EPUB 2.0.1. O padrão EPUB 3.0 é dividido em quatro partes: *EPUB Publications*, *EPUB Content Documents*, *EPUB Open Container Format* e *EPUB Media Overlays*.

O EPUB 3.0 permite criar *e-Books* com o leiaute fluido, ou seja não há a abstração de uma página com um tamanho específico. Assim os livros têm um formato mais fluido, que varia dependendo do dispositivo, e aplicativo de leitura.

Media Overlays permitem realizar o sincronismo do texto e áudio de um e-Book. Essa funcionalidade é baseada no padrão Daisy Talking books (TANK; FREDERIKSEN, 2007) que fornece especificações para criar livros mais acessíveis para pessoas com deficiência visual. Além disso o EPUB3 possui a funcionalidade *text-to-speech* (TTS) que permite síntese de fala. O autor ainda pode controlar as características da fala como pronúncia, fonemas e detalhes da voz que são aplicados a todo o e-Book.

²² sphinx.pocoo.org

²³ http://idpf.org

3.2.2 HTML5

HTML (HICKSON et al., 2014) (*HyperText Markup Language*) é amplamente utilizada como a linguagem de publicação de conteúdo na Web (*World Wide Web*). Ela suporta além de elementos multimídia, linguagens de script e folhas de estilo, formulários para obtenção de dados e objetos embutidos. Em sua recente versão HTML5 (HICKSON et al., 2014) foram inseridos novos recursos multimídia, como por exemplo, a possibilidade de incorporação de vídeo e áudio no documento nativamente.

A linguagem hipermídia HTML foi inicialmente projetada para descrever páginas textuais e por conta disso não possuía forma nativa para representar animações e interações complexas. Para isso normalmente é usada outra tecnologia incorporada ao documento hipermídia, como por exemplo, objetos flash²⁴, linguagens de *script* sendo a mais popular o Javascript (ECMAScript,). Assim, as narrativas virtuais citadas no Capítulo 2 para web são objetos flash incorporados a documentos web.

Com o passar do tempo a linguagem mostra uma clara tendência em separar o conteúdo do documento de seu layout e estilo de apresentação. Isso é feito principalmente através de folhas de estilo, mais comumente escritas em CSS (W3C, 2001) (Cascading Style Sheets). O CSS suporta a definição de estilos para meios e formas de apresentação específicas. Isso é importante para o documento HTML porque permite que ele tenha estilos específicos e distintos para atender a cada situação onde é apresentado. Assim, por exemplo, pode-se criar um estilo específico para impressão do documento e outro para ser apresentado na tela de computadores de forma completamente separada ou compartilhando algumas regras de estilo básicas.

Destaca-se aqui o uso de seletores CSS (W3C, 2011) como forma de relacionar os elementos do documento com os seus estilos de apresentação. Os seletores da linguagem identificam um elemento ou um conjunto de elementos do documento através de suas características. As características que podem ser levadas em conta vão desde identificadores dos elementos até valores de atributos dele.

3.2.3 SMIL

SMIL (Synchronized Multimedia Integration Language) (W3C, 2008) é outra linguagem de autoria hipermídia. A estrutura de seus documentos se assemelha bastante a dos documentos HTML (W3C, 2014). Ela é uma linguagem baseada em XML, modular e é focada na descrição de relacionamentos espacial e temporal de mídias. Ser modular significa que os elementos e componentes da linguagem estão divididos em grupos funcionais com semântica semelhante.

²⁴ <http://www.adobe.com/br/products/flashplayer.html>

Os relacionamentos temporais em SMIL são representados por *containers* temporais. Os principais *containers* são os elementos `< par >` e `< seq >` que são definidos no módulo *BasicTimeContainers*. Outro contêiner temporal da linguagem é o elemento `< excl >` que está contido no módulo *BasicExclTimeContainers*. Cada um desses apresenta uma semântica própria para a apresentação. Para descrever o fluxo da narrativa e os comportamentos temporais mais complexos é utilizada a combinação e composição deles.

O elemento `< par >` tem por significado que todas as mídias e elementos que estejam contidos nele devem ser iniciados simultaneamente. Já o container `< seq >` representa uma relação em que as mídias devem ser apresentadas sequencialmente, ou seja, ao término da primeira mídia do container deve ser apresentada a próxima e assim por diante até que a última mídia encerre a apresentação.

O elemento `< excl >`, por sua vez, apresenta uma semântica semelhante ao do elemento `< par >`, mas com a diferença que somente um dos seus elementos filhos pode estar sendo executado em qualquer dado momento. Assim caso algum de seus elementos filhos inicie a execução, o que está sendo executado no momento deve ser parado.

3.2.4 NCL

NCL (Nested Context Language) (SOARES; RODRIGUES, 2005) é a linguagem hipermídia adotada como o padrão brasileiro para autoria declarativa para TV digital interativa (TVDi) do middleware Ginga-NCL. Essa linguagem é baseada no modelo conceitual NCM (Nested Context Model). Esse modelo define um conjunto de entidades para descrever os nós de mídias e seus relacionamentos de forma separada, abstraindo-se do conteúdo dessas mídias.

Um documento NCL apresenta basicamente duas grandes partes: o cabeçalho do documento `< head >` e o corpo `< body >`. No cabeçalho do documento se encontram informações para descrever layout, características da apresentação, regras de interação entre outras informações que têm por finalidade formatar a apresentação do documento. Já no corpo do documento encontra-se a especificação dos objetos de mídia contidos na apresentação e a definição dos relacionamentos e interações entre eles.

Vale destacar que uma característica importante de NCL é o foco da linguagem em reuso de comportamentos pré-definidos. Isso pode ser notado pelas constantes referências entre as mídias e links presentes no corpo do documento e os elementos definidos em seu cabeçalho.

3.2.5 SceneSync

SceneSync (BUSSON et al., 2016; BUSSON, 2015) é uma linguagem de domínio específico para autoria de objetos de aprendizagem. Ela se baseia no modelo conceitual

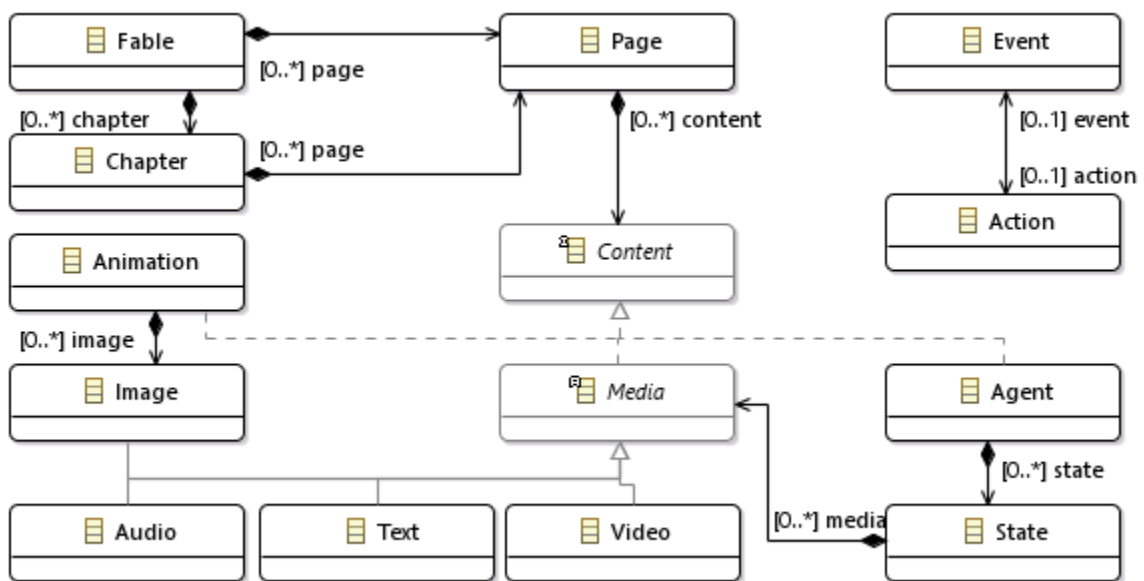
SceneSync Model (SSM). Essa linguagem se baseia fortemente em sincronismo temporal das mídias, controlando a apresentação através da linha temporal. O documento SceneSync usa a abstração de cenas para agrupar os objetos.

As linguagens aqui apresentadas têm um ponto em comum, todas são linguagens modulares. Essa é uma tendência que as linguagens declarativas estão seguindo atualmente. Esses grupos funcionais ou módulos englobam os elementos de acordo com sua funcionalidade de modo que é possível escolher quais funcionalidades se deseja utilizar escolhendo o módulo desejado. Isso permite ainda que as linguagens possam incorporar características e funcionalidades diferentes usando módulos de outra linguagem para uma determinada função.

4 Modelo Conceitual Multimídia Fábulas

O modelo Fábulas é um modelo conceitual multimídia para *e-books* interativos cujos requisitos são discutidos no Capítulo 2. Este modelo visa descrever a forma como esses *e-books* interativos são estruturados. A Figura 18 apresenta uma visão geral do modelo Fábulas.

Figura 18 – Modelo de narrativas do Fábulas



fonte: elaborado pelo(a) autor(a)

A Figura 19 mostra uma representação do modelo conceitual. O conceito de raiz no modelo Fábulas é o *Fable*. O *Fable* representa uma história interativa inteira e é estruturado como um ou mais entidades *Chapters*. Cada *Chapter* pode conter um ou mais *Pages*. O *Fable* também pode não conter nenhum capítulo, caso em que é composto apenas por *Pages* ou uma combinação de ambos. Cada *Page* da história pode ser composta por diferentes objetos de mídia e seu comportamento associado.

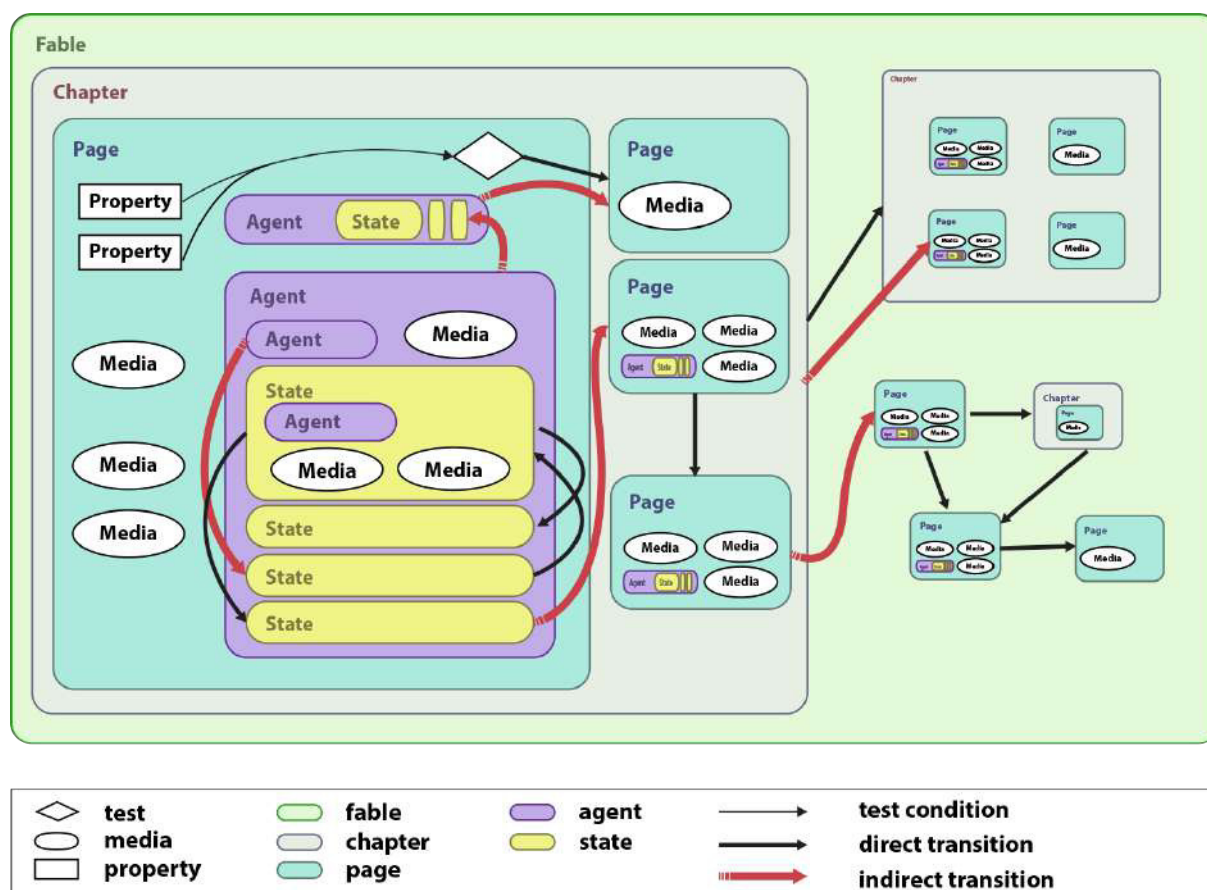
Os objetos de mídia são objetos de mídia comuns encontrados no HTML5, como imagem, vídeo e áudio. Além disso, como discutido na Seção 5.1 o tipo de mídia suportado pode ser estendido através de uma API bem definida. Por exemplo, por padrão, o modelo Fábulas suporta um objeto de mídia chamado *animation* que é composto por uma sequência de imagens que devem ser apresentadas durante uma duração específica.

O comportamento dos objetos de mídia e o controle da história são controlados por um modelo de evento-condição-ação (ECA). A observação desses eventos pode desencadear transições diretas ou indiretas. A transição direta é especificada internamente

no objeto como um agente que muda seu estado. As transições indiretas dependem de eventos desencadeados por outros objetos, tornando-os fracamente acoplados e ajudando a modularizá-los.

Diferente do SMIL, NCL e HTML5, o objetivo foi o design de um modelo mínimo e restritivo para *e-books* interativos infantis. Concebido numa abordagem modular de modo a poder ser facilmente expandido, para fornecer quizzes, jogos, etc.

Figura 19 – Modelo conceitual de um eBook interativo como definido neste trabalho



fonte: elaborado pelo(a) autor(a)

Os *e-books* interativos que seguem o modelo Fábulas podem ser incorporados a outros documentos multimídia, por exemplo páginas Web. A Figura 20 mostra a primeira página da história “O pequeno cavaleiro”.

A Listagem 1 apresenta a primeira página da história “O pequeno cavaleiro”. Esse exemplo é somente a primeira página de uma história completa e ilustra o emprego de diferentes elementos do modelo Fábulas tais como imagem, texto, agentes interativos, e áudio. Pelo fato da máquina de execução do fábulas foi implementada com a abordagem polyfill usando o *framework* AngularJS ¹ é necessário importar a biblioteca, a máquina de execução e o estilo padrão, como visto no exemplo (linhas 6 - 8). Também é preciso

¹ <https://angularjs.org/>

Figura 20 – Exemplo da primeira página do e-book O pequeno cavaleiro.



fonte: elaborado pelo(a) autor(a) com *sprites* retirados do site Game Art 2D <<http://www.gameart2d.com>>

adicionar na página os atributos *ng-app* e *ng-controller* em elementos que iram comportar a raiz < *fable* >, no exemplo foram colocados nos elementos < *html* > e < *body* > respectivamente.

Pode-se ver na página o uso dos atributos *bg-img* e *bg-sound* (line 14) que representam respectivamente, a imagem de fundo e música ambiente da página. Ao se definir uma mídia de texto, se usam as *tags* padrão HTML (HICKSON et al., 2014) que podem ser formatados por folhas de estilo CSS (W3C, 2001). Para posicionar as mídias da página pode-se usar estilos definidos em CSS3 ou os atributos redefinidos pelo modelo Fábulas: “*top*”, “*left*”, “*width*”, “*height*”. O agente (line 17 - 24) tem a função de encapsular a interatividade, nesse exemplo ele representa uma placa que, ao ser tocado (< *on - touch* >) (linhas 20 - 22) emite um evento que encerra a página atual e avança para a próxima página (< *emit* >). As demais páginas da história foram omitidas nesse exemplo.

Listagem 1 – Exemplo de narrativa fábulas em uma página HTML com trecho do código da história “O pequeno cavaleiro”

```

1 <!DOCTYPE html>
2 <html lang="en" ng-app="fablePlayer">
3   <head>
4     <meta charset="UTF-8">
5     <title>O pequeno Cavaleiro</title>
6     <script src="../../src/js/angular.min.js"></script>
7     <script src="../../src/app/fablePlayer.js"></script>
8     <link rel="stylesheet" href="../../src/css/fable-player.css">
9     <link rel="stylesheet" href="little_knight.css">
10  </head>
11  <body ng-controller="fablePlayerController">
12    <fable width="800" height="600">
13      <property name="hasantidote" value="false"></property>
14      <page id="page1" bg-img="../../FablePlayer/Media/BG.png"
15        bg-sound="../../FablePlayer/Media/audio1.mp3">
16        <div class="text" id="title" width="250" height="50" top="50"
17          left="50">O pequeno Cavaleiro</div>
18        
20        <agent id="botao" top="370" right="140" width="105"
21          height="110">
22          <state>
23            
24            <on-touch>
25              <emit event="_END_PAGE"/>
26            </on-touch>
27          </state>
28        </agent>
29
30        <div width="200" height="300" top="100" left="300">
31          <p>Era uma vez um pequeno cavaleiro que recebeu uma carta
32            de um amigo.</p>
33          <p>Ele ficou muito muito surpreso, pois esse amigo
34            tinha sumido a anos.</p>
35          <p>Só que as noticias na carta não eram muito boas. O seu
36            amigo tinha contraído uma doença misteriosa e precisava
37            de ajuda para pegar a cura.</p>
38          <p>O mais rápido que pôde ele foi ao encontro desse
39            amigo.</p>
40        </div>
41      </page>
42    </fable>
43  </body>
44 </html>

```

Nas seções seguintes é explicado com mais detalhes alguns elementos do modelo Fábulas. A Seção 4.1 apresenta os evento e ações usados para controlar o fluxo da navegação. A Seção 4.2 detalha as mídias do modelo Fábulas. A Seção 4.3 apresenta os agentes. A Seção 4.4 apresenta as propriedades e como aplica-las.

4.1 Interação do Usuário: Causalidade e Sincronismo

O fluxo da narrativa pode ser controlado através de um paradigma causal baseado em eventos e ações. As páginas normalmente são apresentadas de acordo com a sequência de definição no *e-book*. Para possibilitar a criação de narrativas não-lineares essa ordem pode ser manipulada através de captura de eventos e definição de transições. A ordem linear da apresentação das páginas por padrão simplifica o desenvolvimento de narrativas lineares, e o autor pode criar desvios no fluxo da apresentação e representar narrativas mais complexas.

4.1.1 Eventos

O evento pode ser dividido em duas categorias: passivos e ativos. Os eventos passivos são acarreitados pelo próprio sistema como o fim de mídias ou alterações de valores em propriedades. Eventos ativos são decorrentes da interação do usuário como, por exemplo, tocar em algum objeto na tela. Para construir narrativas esses dois tipos de eventos podem ser tratados da mesma forma, pois o resultado é sempre o avanço do fluxo da narrativa. A lista de eventos do Fábulas podem ser encontradas na Tabela 10.

Eventos podem ser classificados em três grupos: eventos de apresentação, interação e controle. Eventos de apresentação são aqueles que marcam uma mudança no estado dos elementos da narrativa, por exemplo, o início da apresentação de um áudio ou o fim de um efeito. Evento de interação é um tipo de evento que envolve alguma atividade do usuário ou interação entre os objetos da narrativa, como um toque do usuário em algum objeto ou a colisão entre elementos. Os eventos de controle (*Control*) são usados pelo usuário para poder direcionar o fluxo da narrativa.

O evento `_END_CHAPTER`, `_END_PAGE`, e `_END` determinam, respectivamente, o fim do capítulo, fim da página e fim da narrativa. No caso do fim de um capítulo ou de uma página esse evento indica que a página subsequente deve ser apresentada.

O evento `_PREVIOUS` determina que a página atual deve ser terminada e a que estava sendo exibida anteriormente deve ser reapresentada. O `_PREVIOUS` funciona como uma pilha e salva a ordem em que as páginas foram apresentadas obedecendo até mesmo desvios da narrativa. Na primeira página o evento `_PREVIOUS` não apresenta efeito.

A condição pode ser aplicada em conjunto com a observação do evento para determinar se a comunicação entre os elementos deve prosseguir. As condições são baseadas em valores de propriedades e também podem estabelecer desvios condicionais.

Tabela 10 – Eventos de interação do player

Tipo	Nome	Descrição
Presentation	begin	início da apresentação do objeto
	end	fim da apresentação do objeto
	pause	quando o objeto é pausado
	continue	quando o objeto está em pause e continua do ponto de pausa
	show	quando o componente fica visível
	hide	quando o componente fica invisível
Interaction	touch	quando o usuário toca brevemente o objeto
	hold	quando o usuário toca o objeto e o mantém pressionado por mais de 2s
	release	quando o usuário libera o objeto que estava em estado hold
	drag	quando um objeto em estado hold é movido
	drop	quando o usuário solta um objeto que estava em estado drag
	swipe	quando o usuário movimenta o dedo sobre a superfície da tela
	tilt	quando a tela é girada pelo usuário
	pinch	quando o usuário realiza o movimento de pinch
Control	"name"	evento criado pelo próprio usuário
	_END_CHAPTER	evento que determina o fim de um capítulo
	_END_PAGE	evento que determina o fim de uma página
	_END	evento que determina o fim da narrativa
	_PREVIOUS	evento que determina o retorno a página que estava sendo exibida anteriormente

O elemento origem é quem sofre o evento e o elemento alvo é quem recebe a ação. No caso de eventos de controle o elemento alvo é indeterminado, assim qualquer escopo pode interceptar e tratar esses eventos.

4.1.2 Ações

As ações são usadas para alterar o estado dos objetos da narrativa. Elas podem conter mensagens que são dados complementares da ação. A Tabela 11 apresenta as ações suportadas pelo Fábulas.

A ação *emit* dispara um evento interno para a máquina de execução. Este evento pode ser usado como um gatilho para outras ações em outras partes da fábula. O autor pode informar o escopo de propagação, como um dos valores: *fable*, *chapter* ou *page*. Esses escopos se referem aos elementos *< fable >*, *< chapter >* e *< page >* e todos os seus

Tabela 11 – Ações suportadas pelo Fábulas

Tipo	Nome	Descrição
Presentation	start	inicia a apresentação da mídia
	stop	termina a apresentação da mídia
	pause	pausa a mídia
	resume	continua a apresentação de uma mídia pausada
	show	apresenta o objeto
	hide	esconde o objeto
Control	emit	evoca eventos criados pelo autor
	changeto	transiciona para outro estado ou página
	set	altera o valor de uma propriedade

elementos, que podem então capturar e tratar o evento emitido. Os agentes, mesmo não executando no momento, como os presente em outra página, podem capturar esses eventos.

As ações de *start*, *stop* são usadas para todos os tipos de mídia. *pause* e *resume* correspondem a ações normalmente aplicadas a mídias que têm uma continuidade temporal como vídeos, áudios e animações. Por outro lado, as ações *show* e *hide* visam apenas as mídias gráficas. Elas também podem ocultar mídias temporais como vídeos, sem interromper sua execução. No caso de outros tipos de mídia, como imagens, ele apresenta o mesmo resultado das ações de *start* e *stop*.

4.2 Mídias

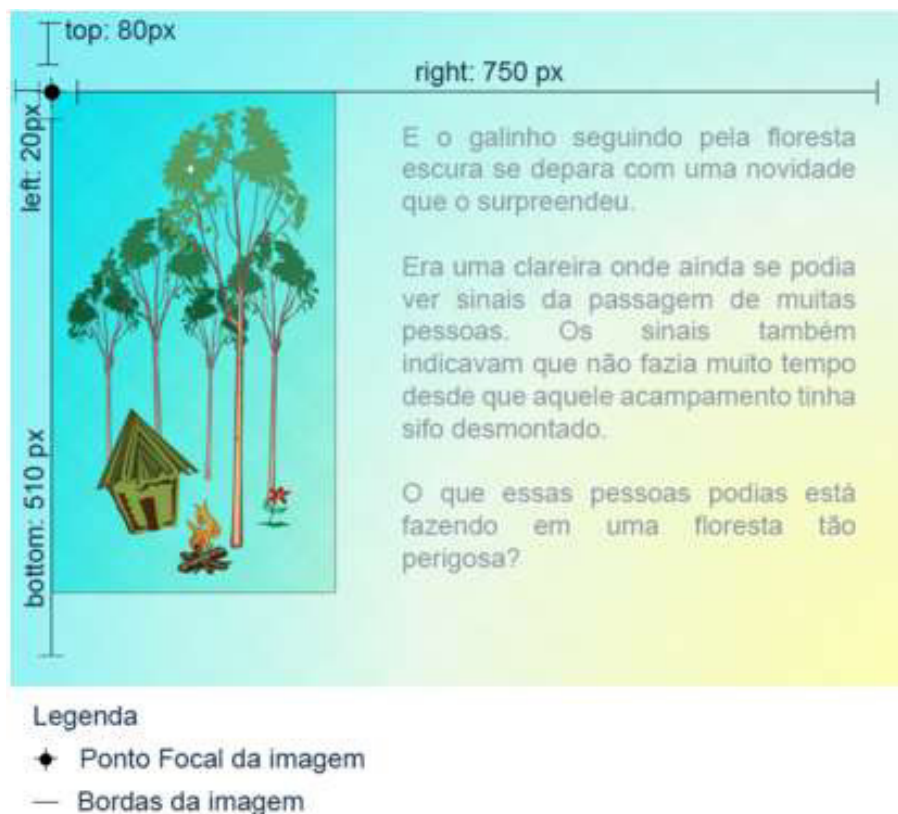
Os *e-books* interativos podem ser descritos em termos de composições multimídias. Essas composições podem ser organizadas pela divisão em páginas e capítulos semelhante a um livro. As páginas compõem a base da estrutura do *e-book* no modelo Fábulas. Os capítulos são estruturas opcionais usadas para agrupar as páginas. Os capítulos também apresentam a função de pontos de referência da narrativa. As mídias que podem compor uma narrativa no modelo Fábulas são imagens, áudios, textos, vídeos, animações, e partículas que são explicados nas subseções a seguir.

4.2.1 Imagem

As imagens podem representar diferentes papéis no modelo Fábulas tais como botões, quadro de uma animação, imagem de fundo etc. As propriedades básicas das imagens são o tamanho e a posição. Para determinar o posicionamento da imagem na cena é usado um esquema de coordenadas relativas. Assim as coordenadas são os valores de distância para as bordas do *container* pai da imagem. Para simplificar o cálculo de uma posição, as coordenadas são determinadas em relação ao canto superior esquerdo

do objeto. A Figura 21 mostra como deve ser feito o posicionamento relativo à imagem selecionada pelo autor.

Figura 21 – Demonstração de posicionamento de uma imagem de acordo com os containers.



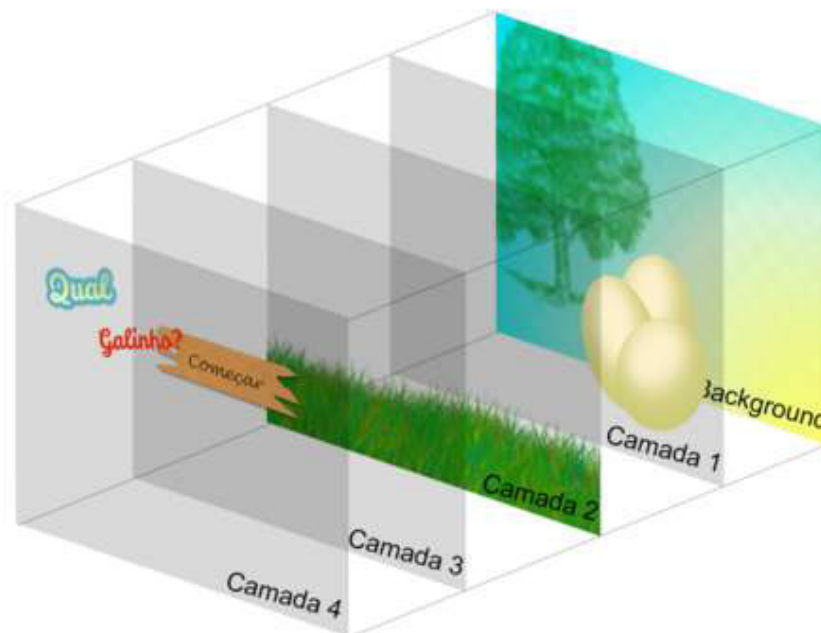
fonte: elaborado pelo(a) autor(a)

Para resolver problemas com sobreposição dos elementos gráficos é utilizada a abstração de camadas (*layers*). As camadas são determinadas automaticamente de acordo com a ordem em que elas são declaradas, onde o último elemento a ser especificado fica na camada mais externa. O autor também pode informar a camada de apresentação de um objeto explicitamente. A Figura 22 mostra como é a organização das camadas que formam a cena.

4.2.2 Texto

O texto precisa ter suas dimensões de exibição e sua posição na cena definida, o que é tratado estabelecendo uma janela de exibição. Todo texto que ultrapassar essa janela de exibição pode ser acessado pelo leitor através da barra de rolagem do texto. Para a formatação textual há a necessidade de exibir o texto em diferentes estilos intercalados no mesmo parágrafo, para isso foi adotada a formatação de estilo em documentos HTML5 (W3C, 2002). Isso permite ao autor aplicar folhas de estilo CSS3 (W3C, 2001) a partes do texto.

Figura 22 – Exemplo de apresentação de uma cena com camadas definidas.



fonte: elaborado pelo(a) autor(a)

4.2.3 Vídeo

As mídias de vídeo no modelo Fábulas seguem a especificação da linguagem HTML5. Elas também obedecem as mesmas regras de posicionamento para as imagens. Entretanto mídias com características temporal como vídeos, áudios e animações devem ser tratadas de modo a atender a necessidade do autor de controlar o início e sua duração. Assim essas mídias fazem uso de evento `<on-end>` e `<on-pause>` de modo mais direto, mesmo que elas não sejam as únicas no modelo Fábulas que usam esses eventos.

O elemento vídeo no modelo Fábulas não pode tratar os seus eventos, logo ele precisa usar um agente para isso. Na Listagem 2 têm-se um exemplo de um vídeo “video1” que tem um agente pareado. Ao terminar a execução do vídeo (`<on-end>`) exibe a imagem “replay.png” para se o leitor desejar repetir o vídeo e quando pausado exibe a imagem “play.png” para continuar o vídeo.

Listagem 2 – Exemplo de vídeo

```
1 <video id="video1" src="video.mp4">
2 </video>
3
4 <agent id="agtVideo">
5   <on-end target="video1">
6     <shout target="agtVideo#end"/>
7   </on-end>
8   <on-pause>
9     <shout target="agtVideo#pause"/>
10  </on-pause>
11  <state id="play">
12    </state>
13  <state id="end">
14    
15    <on-touch>
16      <shout target="agtVideo#play"/>
17      <start target="video1"/>
18    </on-touch>
19  </state>
20  <state id="pause">
21    
22    <on-touch>
23      <shout target="agtVideo#play"/>
24      <restart target="video1"/>
25    </on-touch>
26  </state>
27 </agent>
```

4.2.4 Áudio

O uso de mídias de áudio fornece a possibilidade de enriquecer a experiência do leitor com som ambiente e efeitos sonoros. O som ambiente, por exemplo, pode ser definido separadamente para cada página, como na Listagem 1, ou por capítulo, ou a história toda usando o atributo “*bg-sound*”. O escopo da declaração do som ambiente também influencia, ou seja, caso o som tenha sido declarado para uma página, esta tem prioridade sobre a música declarada para o capítulo que, por sua vez, tem precedência sobre a música do *e-book*. A Listagem 3 mostra um capítulo que tem uma imagem de fundo e um som ambiente declarados.

Listagem 3 – Exemplo de declaração de som ambiente no Fábulas

```

1 <chapter id="chapter1" bg-img="BG.png" bg-sound="audio1.mp3">
2   <page id="page1" bg-sound="audio2.mp3">
3     ...
4   </page>
5   <page id="page2">
6     ...
7   </page>
8   <page id="page3" bg-sound="">
9     ...
10  </page>
11 </chapter>

```

Nesse exemplo, as páginas “page1”, “page2” e “page3” herdam a imagem de fundo “BG.png” do capítulo. A página “page1” declarou a um som ambiente próprio, a música “audio2.mp3”, que tem precedência sobre a música “audio1.mp3” somente nesta página, enquanto a página “page2” continua a herdar o som ambiente “audio1.mp3”. A página “page3” não tem nenhum som ambiente e para anular a herança do atributo ele é declarado vazio.

O áudio declarado diretamente na página é chamado de efeito sonoro e por padrão deve ser ativado em resposta a uma interação do usuário, mas o autor pode usar o atributo *autoplay* para tocar o áudio no momento que ele estiver pronto. Os efeitos sonoros usam o elemento `<audio>` padrão do HTML5. Os efeitos sonoros têm por característica apresentar uma duração limitada e um número de repetições. Um exemplo de efeito sonoro é apresentado na Listagem 4. Nesse exemplo tem-se dois efeitos sonoros: o “openpage” é ativado no momento que a página é iniciada; já o efeito sonoro “bip” é ativado quando o botão “button” é tocado.

Listagem 4 – Exemplo de efeitos sonoros

```

1 <page id="page1">
2   <audio id="openpage" src="audio.mp3" autoplay="autoplay"/>
3
4   <agent id="button">
5     <audio id="bip" src="bip.mp3"/>
6     <state id="state1">
7       
8       <on-touch>
9         <start target="bip"/>
10      </on-touch>
11    </state>
12  </agent>
13 </page>

```

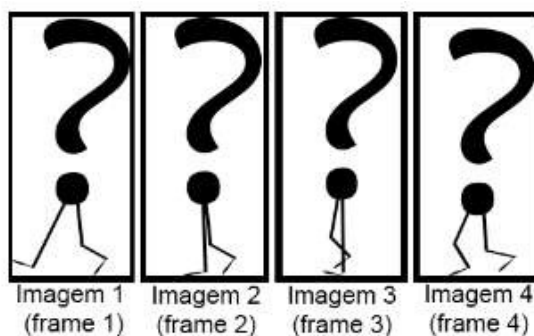

4.2.5 Animação

Para apresentar cenários mais dinâmicos têm-se animações. As animações são divididas em duas categorias principais: animações de *frame* e transições. Cada um desses tipos de animação tem suas próprias particularidades que são discutidas a seguir.

4.2.5.1 Animação de frame

Animação de *frame* pode ser considerada a forma clássica de animação, onde é necessário ter todos os quadros da animação e a troca sucessiva entre esses quadros há a ilusão de movimento. Assim, no modelo Fábulas, esse tipo de animação é tido como um conjunto de imagens sequenciais. A Figura 23 apresenta um exemplo de animação de *frame* como concebido pelo modelo Fábulas.

Figura 23 – Exemplo de imagens de uma animação de *frame*



fonte: elaborado pelo(a) autor(a)

Os atributos usados para possibilitar o controle do comportamento desse tipo de animação são o tempo de duração de cada quadro e o número de repetições da animação. O tempo de duração dos quadros é determinado em milissegundos e é igual para todos os quadros da animação. A sequência de apresentação dos quadros se dá de acordo com a sua declaração no elemento animação. A quantidade de vezes que a animação é apresentada é definida por um número maior ou igual à zero. Caso não seja definido pelo autor ou se for estabelecido que a animação deva repetir infinitamente, o comportamento é que a animação será repetida até que um evento pare sua exibição. Pode-se ver um exemplo de animação na Listagem 5. Ao criar animações, o autor deve ter o cuidado de assegurar que todos os quadros tenham dimensões iguais para evitar deformações ao serem exibidos.

Listagem 5 – Exemplo de animação

```

1 <animation dur="0.3" rep="undefined">
2   
3   
4   
5   
6 </animation>
```

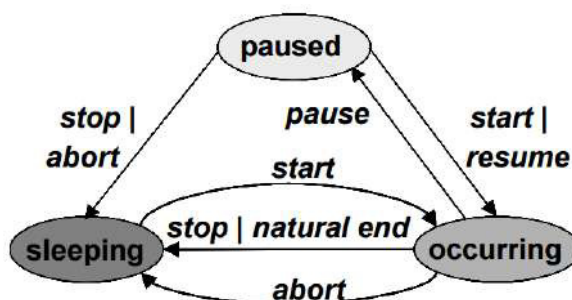
4.2.5.2 Transições

As transições do modelo Fábulas usa a recomendação *Web Animations* da W3C (BIRTLES SHANE STEPHENS, 2016) que define um modelo de sincronização das mudanças de apresentação de uma página web. Essa especificação também define uma API para interagir com esse modelo.

4.3 Agentes

NCM(SOARES; RODRIGUES, 2005) (Nested Context Model) define uma máquina de estados para cada evento. A máquina de estados dos eventos NCM apresenta três estados para cada evento: dormindo (*sleeping*), ocorrendo (*occurring*), e pausado (*paused*). A transição entre os estados obedece a regras que pode ser vista na Figura 24. Cada Evento inicia no estado dormindo e ao inicia a exibição de suas unidades de informação passa ao estado ocorrendo.

Figura 24 – Máquina de estados dos eventos NCM



fonte: extraído de (SOARES; RODRIGUES, 2005)

Entretanto, o modelo NCM permite ao autor somente a manipulação indireta dela através das transições descritas na figura. Agentes (*< agent >*) se baseiam nessa máquina de estados do modelo NCM e podem ser reajustados pelo autor para criar comportamentos mais complexos. O autor pode criar novos estados e definir suas transições no momento da autoria da história usando o elemento *< state >*. Isso não significa que a máquina de estado do NCM é descartada os agentes na verdade estendem o seu comportamento.

No modelo Fábulas, os agentes são um mecanismo que combina o comportamento e a apresentação da mídia. Isso permite que o autor encapsular a semântica do agente em um módulo que pode ser exportado e reusado em diferentes parte da história ou em outras historias sem precisar de grandes alterações. A máquina de estados do agente é definida por uma 5-tupla.

$$SM = \langle S, B, s_0, \Sigma, \delta \rangle$$

Onde:

- S é um conjunto finito e não vazio de estados
- B é a composição base
- s_0 é o estado inicial
- Σ é um conjunto de eventos
- $\delta : S \times \Sigma \rightarrow S$ é a função de transição

Cada elemento da 5-tupla tem suas *tags* correspondentes no modelo Fábulas. A Tabela 12 mostra todas essas equivalências.

Tabela 12 – Equivalências entre máquina de estado de agente e modelo Fábulas

Agente SM	Modelo Fábulas (exemplo)	Descrição
S	<code><state id="S">...</state></code>	definição dos estados filhos do <code>< agent / ></code>
B	<code></code>	mídias ou outros agentes filhos do agente
s_0	<code><state id="s0">...</state></code>	primeiro estado
Σ	<code><on-touch>...</on-touch></code>	eventos filhos de um <code>< agent / ></code> ou <code>< state / ></code>
$\delta : S \times \Sigma \rightarrow S$	<code><changeto target="S" /></code>	filhos de um evento (<code>< on - * / ></code>)

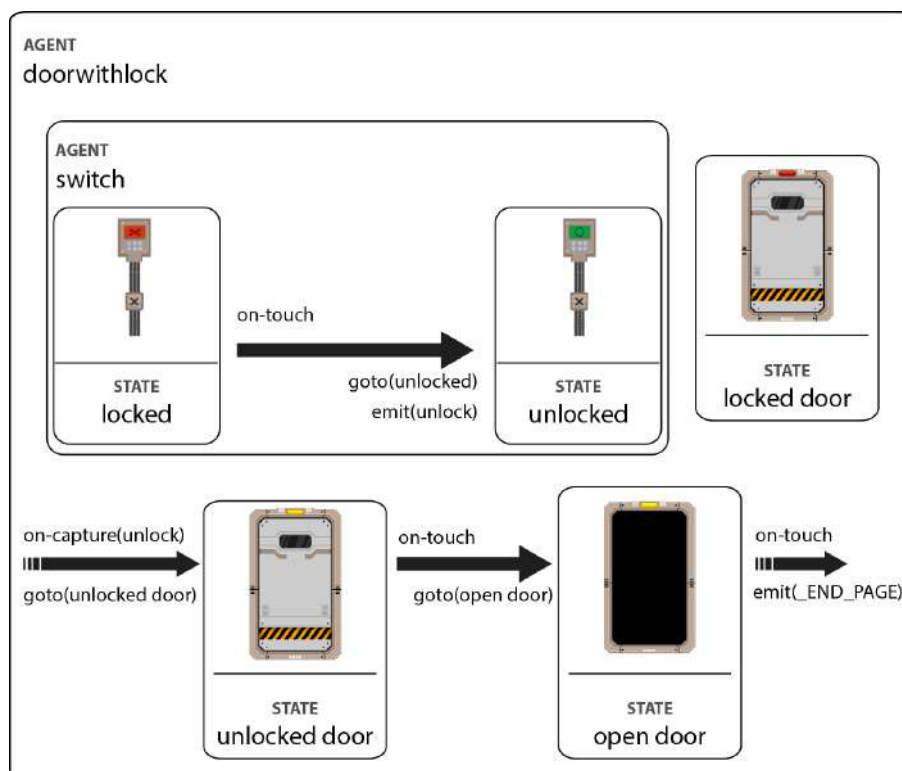
fonte: elaborado pelo(a) autor(a)

O composição base do agente é compostagem dos elementos definidos na sua raiz e em tempo de execução é mesclado com os demais estados. Por padrão, o estado inicial de um agente é o primeiro definido por um elemento `< agent >`, enquanto os estados restantes permanecem suspensos. Um estado pode agrupar objetos de mídia e outros agentes e lidar com eventos emitidos por sua própria composição ou eventos gerados pela interação do usuário.

O estado de um agente pode ser alterado pela ação *changeto* informando o próximo estado. Isso simplifica a criação e exportação de componentes reutilizáveis. A Figura 25 mostra um agente que representa uma porta trancada, e pode ser usado em diferentes situações sem alterações por ser encapsulado.

No exemplo da Listagem 6 tem-se o código que representa o agente da Figura 25. Ele tem um agente (linha 2) que representa o interruptor e três estados. O agente “switch” (linha 2 - 13) e o evento `< on - capture >` (linha 31 - 33) compõem o estado base do agente “doorwithlock”. O agente “switch” contém dois estados “locked” e “unlocked”. Ele começa no estado “locked” e quando tocado ele vai para o estado “unlocked” emitindo um evento personalizado apenas para sua página, chamado “unlock” (linhas 5 - 8). O estado inicial do agente “doorwithlock” é chamado “locked-door” (linhas 15 - 17), que só tem uma imagem. Quando o evento “unlock” é emitido, este agente passa do estado inicial para o estado

Figura 25 – Representação de um agente representado uma porta trancada.



fonte: elaborado pelo(a) autor(a)

“unlocked-door” (linhas 18-23). Quando o usuário toca no agente neste estado, ele vai para o estado “open-door” (linhas 24 - 29). Se o usuário tocar nesse estado, ele encerrará a página com o evento `_END_PAGE` (linha 27).

4.4 Propriedades

As propriedades servem para controle da apresentação ou para representar o comportamento dos objetos. As propriedades são sensíveis ao tipo de objeto ao qual elas pertencem. Por exemplo, objetos sonoros têm a propriedade volume, que os objetos puramente gráficos não têm.

As propriedades de sistema obedecem o escopo das composições, por exemplo, o volume do áudio definido para uma página não interfere no volume das outras páginas. Os tipos de propriedades empregadas no Fábulas são: propriedades de sistema (*system*), propriedades de objeto (*object*) e propriedades de controle (*control*) como apresentado na Tabela 13.

As propriedades de sistema determinam comportamentos globais e variáveis de ambiente, como iluminação do sistema, resolução da tela, volume do aparelho, etc.

As propriedades de objeto são definidas pelos próprios atributos das mídias e objetos que compõe a narrativa. Elas dependem do tipo desses objetos, por exemplo, um

Listagem 6 – Exemplo de agente especificado no modelo Fábulas

```

1  <agent id="doorwithlock">
2    <agent id="switch">
3      <state id="locked">
4        
5        <on-touch>
6          <changeto target="unlocked"/>
7          <emit event="unlock" scope="page"/>
8        </on-touch>
9      </state>
10     <state id="unlocked">
11       
12     </state>
13   </agent>
14
15   <state id="locked-door">
16     
17   </state>
18   <state id="unlocked-door">
19     
20     <on-touch>
21       <changeto target="open-door"/>
22     </on-touch>
23   </state>
24   <state id="open-door">
25     
26     <on-touch>
27       <emit event="_END_PAGE"/>
28     </on-touch>
29   </state>
30
31   <on-capture event="unlock">
32     <changeto target="unlocked-door"/>
33   </on-capture>
34 </agent>

```

objeto gráfico tem largura e altura, diferentemente de um objeto sonoro. Esse tipo de propriedade pode ser simplificado definindo conjuntos de propriedade relacionadas, por exemplo, o tamanho (*size*) definido pela largura e altura.

As propriedades físicas acrescentam características físicas para os objetos. Assim é possível simular peso, colisão e mobilidade para os objetos.

Por sua vez as propriedades de controle são propriedades criadas pelos autores, e funcionam como variáveis. As propriedades de controle precisam obedecer algumas regras de nomenclatura:

- O nome não pode ser nenhuma das propriedades próprias do sistema;
- Não deve haver caracteres especiais;
- Deve-se garantir um nome único.

Tabela 13 – Lista de propriedades do modelo Fábulas

Tipo	Nome	Descrição
<i>system</i>	lightLevel	o nível de inluminação da tela (0 - 1)
	soundLevel	o volume de um objeto sonoro (0 - 1)
	soundLevel	o volume de um objeto sonoro (0 - 1)
<i>object</i>	background	imagem do background
	bgColor	cor do plano de fundo
	backgroundSound	música ambiente
	backgroundSoundRepeat	repetições que a música ambiente
	backgroundSoundPause	intervalo entre as repetições da música ambiente
	left	posicionamento da imagem em relação a borda esquerda
	top	posicionamento da imagem em relação a borda superior
	width	largura da imagem
	height	altura da imagem
	position	combinação do top, e left de um objeto gráfico
	size	combinação do width, height de um objeto gráfico
	bounds	combinação do top, left, width, height de um objeto gráfico
<i>physics</i>	solid	adiciona bordas ao objeto e permite capturar e tratar colisões entre objetos
	heavy	Adiciona peso ao objeto para que este se mova na tela como se fosse pesada
	light	Adiciona peso ao objeto para que este flutue na tela
	draggable	permite que o usuário arraste o objeto na tela
<i>control</i>	“name”	propriedade criada pelo usuário

A Listagem 7 mostra o código de exemplo para propriedade. Nesse código têm-se a imagem “key” que pode ser arrastada pela tela (*draggable*) e a propriedade “hasKey” declarada com o valor *false*. Quando o leitor encontra a chave e a arrasta até o agente “porta” este testa a colisão por serem dois objetos sólidos, ou seja têm a propriedade *solid*, a propriedade “hasKey” é alterada para *true*. Quando o usuário clica na imagem da porta, tal propriedade é testada para confirmar se a chave já foi encontrada.

Listagem 7 – Exemplo de propriedades

```

1 <page id="page2" bg-img="../../FablePlayer/Media/BG.png">
2   <property name="hasKey" value="false"/>
3
4   <agent id="arbusto" top="100" left="300">
5     <state id="closed">
6       
8       <on-touch>
9         <start target="arbusto1#open"></start>
10      </on-touch>
11    </state>
12    <state id="open">
13      
15    </state>
16  </agent>
17  
20    ...
21  <agent solid id="porta" top="300" left="400">
22    <state id="trancada">
23      
25      <on-collision target="key">
26        <set target="hasKey" value="true"/>
27        <stop target="key"/>
28      </on-collision>
29    </state>
30    <state>
31      
33      <on-touch>
34        <test target="hasKey" value="true">
35          <shout target="_END_PAGE"/>
36        </test>
37      </on-touch>
38    </state>
39  </agent>
40 </page>

```

5 Modelando narrativas com o modelo Fábulas

O modelo Fábulas pode ser adaptado para ser usado com diferentes linguagens de autoria multimídia. Na Seção 5.1 é apresentado o uso do Fábulas com HTML em Web Browsers e na Seção 5.2 é apresentado o Fábulas com a linguagem SceneSync.

5.1 Integração com Web Browsers

O modelo discutido acima é integrado na Web usando uma abordagem de polyfill. Para isso, nossa implementação usa AngularJS ¹, Que permite criar facilmente novas *tags* que podem ser integradas em uma página da web. A Tabela 14 apresenta os elementos criados e modificados para permitir executar o modelo Fábulas em web browsers.

AngularJS tem um compilador HTML que permite ao desenvolvedor definir nova sintaxes para elementos HTML ² ou criar novos elementos. Esta abordagem chamada polyfill pode ser útil quando é necessário criar novos elementos declarativos com comportamento personalizado como em DSL e API e também é útil para prototipação rápida pois ele se apoia na estrutura para execução multimídia do navegador somente acrescentando o que é necessário com ajuda de bibliotecas JavaScript. AngularJS foi escolhido pois simplifica esse processo de criação de novos elementos, chamados diretivas. Uma diretiva é definida como um comportamento que deve ser acionado quando construções HTML específicas são encontradas durante o processo de compilação. Essas diretivas podem ser elementos, atributos, nomes de classe, ou comentários. Por exemplo, a Listagem 8 mostra um trecho de código que contém um elemento `<div>` com os atributos do modelo Fábulas *left*, *top*, *width*, *height*. Esses atributos no modelo Fábulas são usados juntamente com os atributos *right* e *bottom* para definir o posicionamento de um elemento, mas eles não são padrão para a linguagem HTML. Assim é necessário definir o comportamento desses atributos, e para isso é criada uma diretiva.

Listagem 8 – Exemplo de código com o uso dos atributos *left*, *top*, *width*, *height*

```

1 <div class="text" id="title" width="250" height="50" top="50"
  left="50">
2   O pequeno Cavaleiro
3 </div>
```

A Listagem 9 mostra o exemplo da definição da diretiva do atributo *left*. O tipo da diretiva (elemento, atributo, nome de classe, ou comentário) pode ser restringido com o

¹ <<https://docs.angularjs.org>>

² <<https://docs.angularjs.org/guide/compiler>>

Tabela 14 – Elementos do modelo Fábulas implementados ou modificados pela máquina de execução

Element	parent	attributes*
fable	-	id, src, width, height, bg-img, bg-sound, bg-sound-rep
chapter	fable	id, src, bg-img, bg-sound, bg-sound-rep, trans-in, trans-out
page	fable, chapter	id, src, bg-img, bg-sound, bg-sound-rep, trans-in, trans-out
property	fable, chapter, page	name, value
img**	page, div, agent, state	left, top, right, bottom, width, height
div**	page, div, agent, state	left, top, right, bottom, width, height
p**	page, div, agent, state	left, top, right, bottom, width, height
span**	page, p, div, agent, state	-
audio**	page, agent, state	-
video**	page, agent, state	left, top, right, bottom, width, height
animation	page, agent, state	id, left, top, right, bottom, width, height
agent	page, agent, state	id, left, top, right, bottom, width, height, draggable, heavy, light, solid
state	agent	id
on-(event)***	fable, chapter, page, agent, state	event****, target, test
(action)***	on-(event)	event****, target, delay, value

* Todos os elementos contem os atributos padrão HTML

** baseados nos elementos HTML e alterados para o modelo Fábulas

*** nome do *event/action*

**** somente para o evento *capture* ou a ação *emit*

parâmetro *restrict* (linha 3) com o valor ‘E’ para definir diretivas de elementos, ‘A’ para atributos, ‘C’ para classes, ‘M’ para comentário ou se necessário qualquer combinação desses valores ‘EA’. No exemplo o código afeta somente os atributos de nome *left*, assim seria possível criar outra diretiva para um elemento ou classe. Logo, ao carregar a página todos os elementos que apresentarem esse atributo terão o comportamento descrito pela função do parâmetro *link* (linha 4). Esse é somente um resumo de como esse framework funciona para os propósitos dessa dissertação.

A Subseção 5.1.1 mostra uma aplicação dessa adaptação do modelo Fábulas com a história “O pequeno cavaleiro”.

Listagem 9 – Definição da diretiva do atributo *left* do modelo Fábulas

```

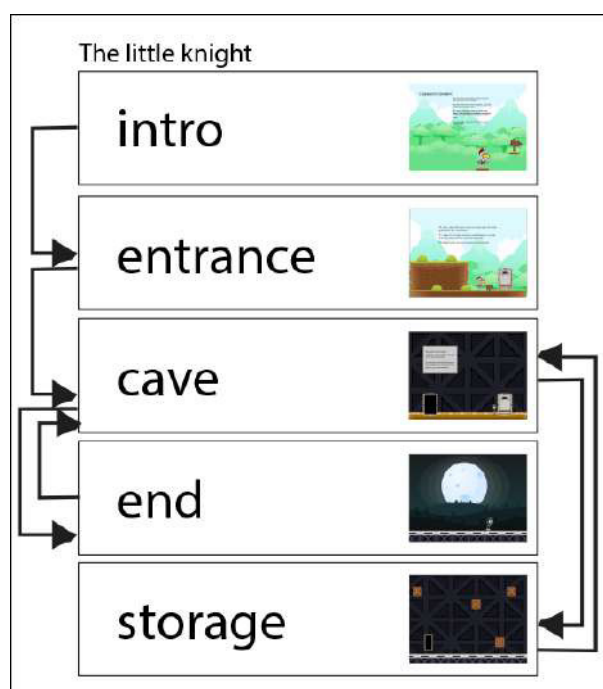
1 fablePlayer.directive('left', ['$document', function($document) {
2   return {
3     restrict: 'A',
4     link: function (scope, element, attr) {
5       element.css({
6         left: attr.left + 'px'
7       });
8     }
9   };
10 }]);

```

5.1.1 Exemplo de *e-book*: O pequeno cavaleiro

Esse e-book é uma história curta de um cavaleiro que ao receber notícias de um amigo, com o qual há muito tempo havia perdido contato, parte em uma jornada para tentar ajudá-lo. A história se desenrola em cinco páginas, destas quatro fazem parte da história principal e a última é um desvio da narrativa. A Figura 5.1.1 apresenta o esquema de navegação do e-book interativo “O pequeno cavaleiro”.

Figura 26 – Diagrama de navegação entre as páginas do e-book interativo “O pequeno cavaleiro”



fonte: elaborado pelo(a) autor(a)

A primeira página dessa história é apresentada no início do Capítulo 4 na Listagem 1. As próximas listagens exploram a segunda página da história.

A Listagem 10 mostra os elementos estáticos da página “entrance”. Esses elementos formam o cenário (linhas 8 - 10) e o texto (linhas 2 - 6) a ser apresentado na tela.

Listagem 10 – Código parcial da página “entrance” da história “O pequeno cavaleiro” somente com os elementos estáticos

```
1 <page id="entrance">
2   <div left="200" width="400" top="100">
3     <p>Na carta o amigo falou que estava na caverna perto da
4       cidade onde achava que a cura estava.</p>
5     <p>Ao chegar lá o cavaleiro notou que estranhamente a caverna
6       tinha uma porta metálica e que estava trancada.</p>
7     <p>Procurando pistas ele notou uma placa perto da porta.</p>
8   </div>
9
10  
12  
14  
16  ...
17 </page>
```

Nessa página também há uma placa do lado da porta que ao ser tocada mostra um aviso para o leitor onde ele deve encontrar uma chave para abrir a porta. A Listagem 11 mostra o código para o agente dessa placa. Nas linhas 3 - 6 temos uma animação simples da placa enquanto ela está no estado (*< state >*) “rebolando”. Caso seja tocada nesse estado (*< on - touch >*) ela muda para o estado “aviso”. Após três segundos, por causa do *delay* na ação *changeto* (linha 9) ou quando tocada (linhas 17 - 18) novamente ela retorna para “rebolando”.

Listagem 11 – Código da placa interativa da página “page2”

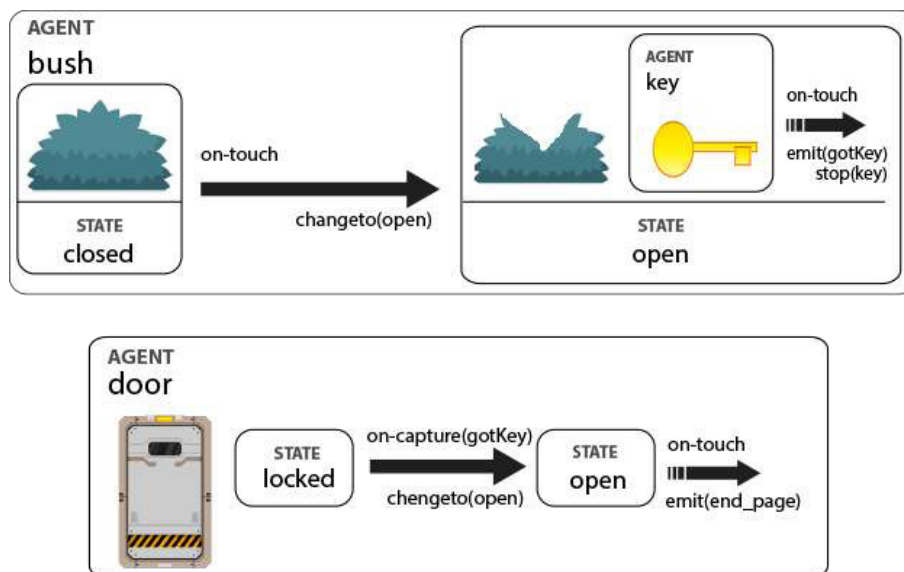
```

1 <agent id="placa" top="510" right="230" height="50" width="50">
2   <state id="rebolando">
3     <animation dur="0.5" rep="indefined">
4       
6       
8     </animation>
9     <on-touch>
10      <changeto target="aviso"/>
11      <changeto delay="3" target="rebolando"/>
12    </on-touch>
13  </state>
14  <state id="aviso">
15    
16    <div class="aviso">
17      A chave está nos arbustos
18    </div>
19    <on-touch>
20      <changeto target="rebolando"/>
21    </on-touch>
22  </state>
23 </agent>

```

A página “entrance” tem quatro arbustos um com a chave e três vazios. A Figura 5.1.1 mostra esquematicamente o comportamento desse agente. A Listagem 12 mostra o código do arbusto com a chave. A diferença desse arbusto pros outros é a posição e o agente “key” (linhas 10 - 18) presente no estado “open” do agente. O agente “key” ao ser tocado emite o evento “gotkey” que é então capturado pela porta.

Figura 27 – Agente “bush” e agente “key” interagindo com o agente “door”.



fonte: elaborado pelo(a) autor(a)

Listagem 12 – Código do arbusto com a chave da página “page2”

```

1 <agent id="arbusto" top="310" left="50">
2   <state id="closed">
3     
4     <on-touch>
5       <changeto target="open"/>
6     </on-touch>
7   </state>
8   <state id="open">
9     
11     <agent id="key" top="0" left="0">
12       <state>
13         
15         <on-touch>
16           <emit event="gotkey" scope="page"/>
17           <stop target="key"/>
18         </on-touch>
19       </state>
20     </agent>
  </state>
</agent>

```

A Listagem 13 mostra o código da porta da página “entrance”. Nas linhas 4 - 6 tem-se o evento *on-capture* quando o agente emite o evento “gotkey” muda para o estado “unlocked”. Quando no estado “unlocked”, ao ser tocado (< *on - touch* >) a porta emite o evento *_END_PAGE* que informa que a apresentação deve encerrar a página atual e iniciar a próxima.

Listagem 13 – Código da porta da página “page2”

```

1 <agent id="door" bottom="50" right="100" height="150" width="100">
2   
4   <state id="locked">
5     <on-capture event="gotkey">
6       <changeto target="unlocked"/>
7     </on-capture>
8   </state>
9   <state id="unlocked">
10    <on-touch test="hasKey">
11      <emit event="_END_PAGE"/>
12    </on-touch>
13  </state>
</agent>

```

Na página “cave” é possível seguir dois caminhos o primeiro para a página “storage” e o outro para a página final “end”. Nessa página tem duas portas e dependendo da porta que o leitor escolher ele segue para uma sala ou outra. A Listagem 14 apresenta os agentes responsáveis por isso. O evento “armazem” (linha 7) é capturado pela página (linhas 37 - 39) que muda para página “storage”. O agente “portafinal” (linhas 11 - 35) emite o evento

`_END_PAGE` (linha 29) que encerra a página atual e segue para a página na sequência, “end”.

Listagem 14 – Código resumido da página “page3”

```
1 <page id="storage" bg-img="../../FablePlayer/Media/bg3.png">
2 ...
3   <agent id="portaArmazem" bottom="30" left="100" height="150"
4     width="100">
5     <state>
6       
8       <on-touch>
9         <emit event="armazem" scope="page"/>
10      </on-touch>
11    </state>
12  </agent>
13  <agent id="portafinal" bottom="30" right="100" height="150"
14    width="100">
15    
17    <agent id="switch">
18      
20      <state id="locked">
21        <on-touch>
22          <emit event="unlock" target="page"/>
23          <changeto target="unlocked"/>
24        </on-touch>
25      </state>
26      <state id="unlocked">
27      </state>
28    </agent>
29    <state id="trancada">
30    </state>
31    <state id="aberta">
32      
34      <on-touch>
35        <emit event="_END_PAGE"/>
36      </on-touch>
37    </state>
38    <on-capture event="unlock">
39      <changeto target="aberta"/>
40    </on-capture>
41  </agent>
42  <on-capture event="armazem">
43    <changeto target="storage"/>
44  </on-capture>
45 </page>
```

Na página “storage” - cujo código fonte não é mostrado aqui principalmente para brevidade, uma vez que, em princípio, utiliza estruturas semelhantes já às páginas já

apresentadas — o leitor pode encontrar o antídoto para o amigo do cavaleiro em um dos três "box"agentes. Esses agentes funcionam como os agente da “bush” da Figura 5.1.1. Quando o usuário encontra esse antídoto, ele altera a variável “hasAntidote” para *true*. Na página “end”, a variável “hasAntidote” é testada, e se o seu valor for *true*, então, o amigo do cavaleiro será curado, e a história termina; Caso contrário, o leitor tem que voltar a procurar o antídoto na página “storage”.

5.2 SceneSync

Esta seção apresenta a extensão da linguagem SceneSync em conformidade com o modelo Fábulas. A linguagem SceneSync já possui um modelo de sincronismo temporal conciso e robusto. Ela foi projetada para ser modular, assim facilitando a sua adaptação ao modelo conceitual do Fábulas. Para essa adaptação somente se mostra necessário acrescentar módulos responsáveis por animações, interação com usuário, e componentes. A Tabela 5.2 lista os novos elementos acrescentado ao SceneSync.

Tabela 15 – Elementos do módulo Interaction

Elemento	Atributos	Elementos filhos
select	id, target, value	(start, stop, set, goto)*
case	id, target, value	(start, stop, set, goto)*
property	name, value	-
component	id	(state)*
state	id	(image, text, audio, video, scene)*
animation	id, dur, repeat	image*

fonte:elaborado pelo(a) autor(a)

A narrativa escolhida de exemplo se chama “Ajude o Pirata”, que é uma história interativa sobre um pirata que precisa de ajuda para encontrar uma chave para abrir um baú do tesouro, a Figura 28 apresenta as duas primeiras cenas da história. A narrativa é composta por três cenas. Na primeira cena, o pirata é apresentado e pede ajuda ao leitor para encontrar a chave que abre o baú. Na segunda cena, é exibida uma ilha onde o usuário deve procurar pela chave perdida. E na terceira cena, o leitor é parabenizado ao encontrar a chave.

Para cada cena é definido um identificador único e as suas respectivas mídias. Também é necessária a definição de uma variável global “HoldKey” que indica se leitor já encontrou a chave. O componente “bush”, definido na segunda cena, representa um

Figura 28 – Cenas da narrativa “Ajude o Pirata”



fonte: elaborado pelo(a) autor(a)

arbusto e possui dois estados, no primeiro ele exibe a imagem de um arbusto fechado, e no segundo, uma imagem de um arbusto aberto.

Na primeira cena, a narrativa deve iniciar exibindo as imagens que a compõe. Para isso, conforme ilustra as linhas 1 a 6 da Listagem 15, define-se um objeto de sincronismo (*< sync >*) com objetos de ação do tipo *< start >*, que iniciam a apresentação das imagens de fundo, pirata, balão de fala e baú do tesouro.

No balão de fala do pirata é informado que, ao tocar a tela do dispositivo, é iniciada a busca pela chave perdida. Para isso, conforme ilustrado nas linhas 7 a 9 da Listagem 15, define-se um objeto de seleção (*< select >*) com o atributo *event* definido como “touch”. Dentro do objeto de seleção é definido um objeto de ação do tipo *< goto >* que redireciona a aplicação para a cena “scene_island”. Dessa forma, ao tocar na tela, a aplicação deve iniciar a apresentação da cena da ilha.

Como feito na primeira cena, também se deve iniciar as mídias que compõe a segunda cena. Para isso, conforme ilustra a Listagem 15, é declarado um objeto de sincronismo que contém as ações do tipo *< start >* que iniciam as imagens do plano de fundo (areia, palmeira, mar e pedra), bem como o componente “bush”. Esse componente tem como primeiro estado a exibição da imagem do arbusto fechado. Note que a imagem da chave não foi iniciada, esta deve ser exibida apenas quando o leitor encontrar sua localização.

Listagem 15 – Conteúdo das cenas da narrativa.

```

1 <scenesync>
2   <var id="HoldKey" value="false" />
3
4   <scene id="scene_pirate">
5     <image src="balao_fala.png" id="speaks" />
6     <image src="pirate.png" id="pirate" />
7     <image src="back_pirate.png" id="back_pirate" />
8     <image src="bau.png" id="chest" />
9     <sync>
10      <start target="speaks" />

```



```

11     <start target="pirate" />
12     <start target="back_pirate" />
13     <start target="chest" />
14 </sync>
15 <select event="touch">
16     <goto target="scene_island" />
17 </select>
18 </scene>
19 <scene id="scene_island">
20     <image src="back_pirate.png" id="back_island" />
21     <image src="areia.png" id="sand" />
22     <image src="mar.png" id="sea" />
23     <image src="palmeira.png" id="palmtree" />
24     <image src="pedra.png" id="rock" />
25     <image src="key.png" id="key" />
26 <sync>
27     <start target="sand" />
28     <start target="sea" />
29     <start target="palmtree" />
30     <start target="rock" />
31 </sync>
32 <component id="bush">
33     <state id="closed">
34         <image src="arbusto.png" id="bush_closed" />
35         <select event="touch">
36             <start target="opened" />
37             <case target="HoldKey" value="false">
38                 <start target="key" />
39             </case>
40         </select>
41     </state>
42     <state id="opened">
43         <image src="arbusto_aberto.png" />
44     </state>
45 </component>
46 </scene>
47 <scene id="scene_congrats"> ... </scene>
48 </scenesync>

```

Na narrativa, a chave perdida está escondida dentro do arbusto. Logo, quando o usuário tocar o arbusto, este deve trocar seu estado para “bush_open” e a imagem da chave deve ser exibida. Para isso, conforme ilustra a Listagem 1, define-se dentro do objeto de imagem “bush_closed” um objeto de seleção com o atributo *event* definido como *touch*. Dentro dele, um objeto de ação do tipo *< start >* que inicia o estado “opened” do componente “bush”. Dentro do objeto de seleção também há uma estrutura de condição que verifica se o valor da variável “HoldKey” é falso. Isto é, verifica se o leitor não segura a chave. Caso a condição seja satisfeita, o objeto de ação do tipo *< start >* inicia a apresentação da imagem da chave.

6 Conclusões

Este trabalho trata da engenharia de documentos para a plataforma Fábulas para autoria de *e-books* interativos. Como principais contribuições desse trabalho têm-se:

- **Modelo conceitual Fábulas:** O Capítulo 4 discute o modelo Fábulas. Esse modelo conceitual é a principal contribuição desse trabalho. Ele consolida toda a pesquisa apresentada nessa dissertação em definições de processos e regras de como se deve criar *e-books* interativos.
- **Requisitos de *e-books* infantis interativos:** O Capítulo 2 apresenta os requisitos do ponto de vista de desenvolvedores para *e-books* interativos que combinados com os requisitos apresentados por Cruz (2016) podem ser usados para guiar o desenvolvimento de ferramentas de autoria para esse domínio.
- **Levantamento de ferramentas de autoria e linguagens multimídia:** O Capítulo 3 discute as diferentes abordagens que podem ser assumidas para autoria multimídia, focando especialmente em discutir o domínio de *e-books* interativos.

O modelo conceitual Fábulas se mostra simples e abrangente. No Capítulo 5 mostra como esse modelo é versátil ao apresentar a aplicação dele com duas linguagens multimídias diferentes HTML e SceneSync. Ao invés de optar pela criação de mais uma nova linguagem, neste trabalho é apresentado como especificar extensões específicas para o uso do Fábulas. Essas extensões são usadas para dar ao modelo conceitual SceneSync e navegadores web a capacidade de expressar narrativas interativas.

O trabalho usa a modelagem de exemplos de *e-books* interativos para validar a proposta. Essas aplicações mostram que o modelo Fábulas foi satisfatório para expressar tal exemplo de aplicação multimídia com concisão e simplicidade.

Para a realização desse trabalho foi realizado um levantamento de requisitos para autoria de *e-books* interativos infantis. Esse conhecimento de domínio define, portanto, as diretrizes que regem a concepção do modelo conceitual Fábulas.

6.1 Trabalhos Futuros

O modelo Fábulas é modular e pode ser facilmente expandido. Alguns pontos que foram discutidos no levantamento de requisitos e não foram contemplados nessa versão inicial do modelo podem ser acrescentado no futuro. Desses pontos vale destacar:

- **Narração:** Embora seja possível simular a narração através de mídias de áudio. É desejável que esta funcionalidade seja oferecida de uma forma mais direta e simplificada pelo modelo. O **Highlight de texto** é uma funcionalidade muito aplicada nesses aplicativos que também seria interessante ser abordado pelo modelo.
- **Múltiplas linguagens:** O modelo precisa fornecer suporte a múltiplos idiomas para as histórias.
- **Mais animações e físicas:** Embora haja o suporte para animações simples, mas animações mais complexas como partículas, filtros em imagens etc, ainda não são possíveis com essa versão do Fábulas. Físicas mais complexas como juntas e molas também seriam interessantes adições para o Fábulas.
- **Biblioteca de documentos:** Essa funcionalidade já está planejada para o modelo mas ainda não foi implementada. Assim seria possível por exemplo definir bibliotecas de mídias ou elementos e importa-los em múltiplas histórias.

Uma extensão do Fábulas para *e-books* EPUB também está em estudo. Essa extensão é semelhante a apresentada para navegadores web, já que EPUB é baseado em HTML. Contudo é necessário discutir alguns problemas como: salvar estados entre páginas; integrar as páginas Fábulas na tabela de navegação do EPUB; garantir compatibilidade com os leitores de EPUB disponíveis.

Outro trabalhos futuro interessantes seria pensar no emprego de engenharia dirigida por modelos (MDE) para a especificação do meta-modelo que descreve o modelo conceitual multimídia Fábulas e realizar a transformação entre as diferentes linguagens de autoria multimídia. Também pode-se pensar em como formalizar a especificação de uma Linguagem de Domínio Específico (DSL) visual para Fábulas.

6.2 Trabalhos Publicados

Esta dissertação teve um trabalho aceito para apresentação como artigo curto no 22º Simpósio Brasileiro de Multimídia e Web ¹. O trabalho é intitulado “Creating non-linear interactive narratives with Fábulas model” e discute a aplicação do modelo Fábulas com o modelo conceitual SceneSync.

Outro trabalho está sendo escrito e será submetido para o evento 17th ACM Symposium on Document Engineering (DocEng 2017) ². O trabalho tem o título “The Fábulas Model for Authoring Web-based Children’s eBooks” e discute a aplicação do modelo Fábulas em navegadores web.

¹ <<https://www5.ifpi.edu.br/webmedia/>>

² <<https://doceng.org/newsite/doceng2017>>

Referências

- ALBANESIUS, C. Apple targets educators via ibooks 2, ibooks author, itunes u app. *PC magazine*, 2012. Disponível em: <<http://www.pcmag.com/article2/0,2817,2399075,00.asp>>. Acesso em: 22 fev. 2017. Citado na página 33.
- BENEDETTI, S. M. Ebook interativo: hipermídia no livro eletrônico. 2016. Citado na página 14.
- BIRTLES SHANE STEPHENS, A. D. T. A. B. *Web Animations*. 2016. <<https://www.w3.org/TR/2016/WD-web-animations-1-20160913/>>. Accessed: 2017-03-07. Citado na página 58.
- BJARNASON, B. The ibooks 2.0 textbook format. *Baldur B/arnason, January*, v. 19, 2012. Citado na página 33.
- BOTT, E. How apple is sabotaging an open standard for digital books. *Bott Report (blog). ZDNet, January*, v. 22, 2012. Citado na página 33.
- BUSSON, A. J. G. *SceneSync: Uma Linguagem de Autoria Hipermídia para Descrição de Objetos de Aprendizagem*. Dissertação (Mestrado) — Universidade Federal do Maranhão, São Luis, Brasil, 2015. Citado 2 vezes nas páginas 16 e 44.
- BUSSON, A. J. G. et al. Scenesync: A hypermedia authoring language for temporal synchronism of learning objects. p. 175–182, 2016. Citado 2 vezes nas páginas 16 e 44.
- CONBOY, G. et al. *EPUB 3 Overview. IDPF Recommended Specification 11 October 2011*. 2011. Citado 3 vezes nas páginas 14, 35 e 42.
- CRUZ, P. T. M. B. da. *Padrões de Interação para Ferramentas de Autoria de Storytellings Infantis*. Dissertação (Mestrado) — Universidade Federal do Maranhão, São Luis, Brasil, 2016. Citado 4 vezes nas páginas 17, 19, 31 e 74.
- DEURSEN, A. V.; KLINT, P.; VISSER, J. Domain-specific languages: An annotated bibliography. *Sigplan Notices*, v. 35, n. 6, p. 26–36, 2000. Citado na página 15.
- EBNER, M. et al. Potential of epub3 for digital textbooks in higher education. *arXiv preprint arXiv:1606.05690*, 2016. Citado na página 17.
- ECMASCRIPT, E.; ASSOCIATION, E. C. M. et al. *Ecmascript language specification*. 2016. <<https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>>. Accessed: 2017-03-07. Citado 2 vezes nas páginas 36 e 42.
- ECMASCRIPT, I. Iso/iec 16262: 19 98. *ECMAScript Language Specification n. Available from ECMA (European Computer Manufacturers Association) at http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM*. Citado na página 43.
- GLAZMAN, D. ibooks author, a nice tool but. *Personal web blog. Retrieved October*, v. 5, 2012. Citado na página 33.
- HART, M. *Project gutenber*. [S.l.]: Project Gutenberg, 1971. Citado na página 14.

- HICKSON, I. et al. *Html5. w3c recommendation. World Wide Web Consortium (W3C)*, 2014. Citado 6 vezes nas páginas 16, 33, 36, 42, 43 e 48.
- INCORPORATED, A. S. *PDF Reference*. 2006. <http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf>. Accessed: 2017-03-07. Citado na página 35.
- INCORPORATED, A. S. *PDF Reference, sixth edition: Adobe Portable Document Format version 1.7*. 2006. Disponível em: <http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf>. Citado na página 14.
- LEBERT, M. *A short history of ebooks*. 2009. Citado na página 14.
- MANA, N. et al. *Interactive e-books for children*. p. 593–595, 2013. Citado na página 17.
- MILLER, B. N.; RANUM, D. L. *Beyond pdf and epub: toward an interactive textbook*. p. 150–155, 2012. Citado na página 42.
- MOBI. 2016. <https://wiki.mobileread.com/wiki/MOBI>. Accessed: 2017-04-10. Citado na página 14.
- MORI, K. et al. *Interactive rich reading: enhanced book reading experience with a conversational agent*. p. 825–826, 2011. Citado na página 34.
- SHEET, S. *itunes 2, apple computer. Inc., Oct*, v. 31, 2001. Citado na página 14.
- SIGARCHIAN, H. G. et al. *Towards cost-effective enrichment of epub3-compliant ebooks. Ghent University, Korea Advanced Institute of Science and Technology*, 2013. Citado na página 35.
- SOARES, L. F. G.; RODRIGUES, R. F. *Nested context model 3.0: Part 1—ncm core. Monografias em Ciência da Computação do Departamento de Informática, PUC-Rio*, n. 18/05, 2005. Citado 2 vezes nas páginas 44 e 58.
- TANK, E.; FREDERIKSEN, C. *The daisy standard: entering the global virtual library. Library trends*, The Johns Hopkins University Press, v. 55, n. 4, p. 932–949, 2007. Citado na página 42.
- W3C. *Introduction to CSS3*. 2001. <<https://www.w3.org/TR/css3-selectors/>>. Acessado em: 31/05/2016. Citado 5 vezes nas páginas 33, 36, 43, 48 e 53.
- W3C. *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)*. 2002. <<http://www.w3.org/TR/2002/REC-xhtml1-20020801/>>. Acesso em 20 de janeiro de 2014. Citado 2 vezes nas páginas 42 e 53.
- W3C. *Synchronized Multimedia Integration Language (SMIL 3.0) Specification*. 2008. <<http://www.w3.org/TR/2008/REC-SMIL3-20081201/>>. Acesso em 20 de janeiro de 2014. Citado na página 43.
- W3C. *Selectors Level 3*. 2011. <<https://www.w3.org/TR/css3-selectors/>>. Acessado em: 31/05/2016. Citado na página 43.
- W3C. *HTML5 Specification*. 2014. <<https://www.w3.org/TR/html5/>>. Acessado em: 31/05/2016. Citado na página 43.

WIEGAND, T. et al. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, IEEE, v. 13, n. 7, p. 560–576, 2003. Citado na página [34](#).