INSTITUTO FEDERAL
SANTA CATARINA

CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS

CURSO SUPERIOR DE TECNOLOGIA EM
GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

GABRIELA ANDREA SOREL

Modelo de Aprendizagem Lúdica de XML Baseado em um Jogo de Computador

Florianópolis - SC
2017

Ficha de identificação da obra elaborada pelo autor

Sorel, Gabriela Andrea
Modelo de Aprendizagem Lúdica de XML baseado em um Jogo de Computador / Gabriela Andrea Sorel ; orientação de Gilmar Carvalho de Souza; coorientação de Underlea Cabreira Corrêa. - Florianópolis, SC, 2017.
69 p.

Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. CST em Gestão de TI. Departamento Acadêmico de Saúde e Serviços.
Inclui Referências.

1. Aprendizagem Lúdica. 2. XML. 3. Jogos de Computador. 4. Educação. 5. Serious Games. I. Carvalho de Souza, Gilmar. II. Cabreira Corrêa, Underléa. III. Instituto Federal de Santa Catarina. Departamento Acadêmico de Saúde e Serviços. IV. Título.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA

DEPARTAMENTO ACADÊMICO DE SAÚDE E SERVIÇOS
CURSO SUPERIOR DE TECNOLOGIA EM GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

GABRIELA ANDREA SOREL

MODELO DE APRENDIZAGEM LÚDICA DE XML BASEADO EM UM JOGO DE COMPUTADOR

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Tecnólogo em Gestão da Tecnologia da Informação.

Orientador:
Prof. Gilmar Carvalho de Souza, Meng.

Coorientadora:
Profª. Underlea Corrêa, Dra.

FLORIANÓPOLIS - SC
DEZEMBRO/2017

MODELO DE APRENDIZAGEM LÚDICA DE XML BASEADO EM UM JOGO DE COMPUTADOR

GABRIELA ANDREA SOREL

Este trabalho foi julgado adequado para obtenção do Título de Tecnólogo em Gestão da Tecnologia da Informação e aprovado na sua forma final pela banca examinadora do Curso Superior de Tecnologia em Gestão da Tecnologia da Informação do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis-SC, 15 de novembro de 2017.

Prof. Julio Cesar da Costa Ribas, Dr.
Coordenador do CST em Gestão da Tecnologia da Informação
Instituto Federal de Santa Catarina

Banca Examinadora:

Prof. Gilmar Carvalho de Souza, MEng
Orientador
Instituto Federal de Santa Catarina

Herval Daminelli

Prof. Herval Daminelli, Esp.
Instituto Federal de Santa Catarina

Adriano Heis

Prof. Adriano Heis, Msc
Instituto Federal de Santa Catarina

Às crianças, por uma aprendizagem mais lúdica

AGRADECIMENTOS

Agradeço, primeiramente, ao Professor Gilmar, o meu orientador paciente e dedicado. Sem sua importante participação este trabalho não teria sido possível!

Também agradeço ao John, meu companheiro de caminhada, que me incentivou desde o começo e ajudou na criação de algumas imagens.

Finalmente, mas não menos importante, agradeço ao Juan, meu filho amado, pela paciência nas horas em que estive ausente.

Uma menção especial para o IFSC-Florianópolis. Muito do que se fala por aí da Escola Pública no Brasil não é verdade. A verdade é que temos excelentes professores, com vocação e formação de qualidade, e cursos atualizados com estrutura suficiente para termos a melhor experiência de ensino-aprendizagem. Só temos que lutar contra a mercantilização da educação no país. Só mantendo escolas públicas, gratuitas e de qualidade, é que todos teremos as mesmas oportunidades. Por um país mais justo, apoiemos o ensino público. Exijamos dos governos mais investimentos em educação!

“É no brincar, e somente no brincar, que o indivíduo, criança ou adulto, pode ser criativo e utilizar sua personalidade integral; e é somente sendo criativo que o indivíduo descobre o eu (self)”.

WINNICOTT, Psicanalista.

“Uma das funções básicas do brincar é permitir que a criança aprenda a elaborar/resolver situações que vivencia no seu dia-a-dia”.

VYGOTSKY, Psicólogo.

“Não eduques as crianças nas várias disciplinas recorrendo à força, mas como se fosse um jogo, para que também possas observar melhor qual a disposição natural de cada um”.

PLATÃO, Filósofo.

RESUMO

No presente trabalho foi realizada uma pesquisa aplicada qualitativa para propor um modelo de aprendizagem lúdica da linguagem de marcação XML baseado em jogos de computador. Na primeira fase, a pesquisa foi orientada a analisar como ocorre a aprendizagem e conhecer alguns dos jogos e métodos existentes para aprendizagem de linguagens de computador. Na sequência, usando o software Netbeans, foi criado o jogo propriamente dito, o qual foi chamado de XML-KIDS. O jogo foi descrito em linguagem UML com os casos de uso e seus diagramas de classe e de sequência. Finalmente foram apresentadas as conclusões e algumas ideias para trabalhos futuros.

Palavras-chave: Aprendizagem lúdica. XML. Jogos de computador. Educação.

ABSTRACT

In this work, a qualitative applied research was accomplished to offer a model for learning XML in a ludic manner, based on a computer game – a serious game. Initially, the research aimed to analyze how learning process occurs, and to meet some of the existing games and methods designed for helping in computer languages learning process. Then, using Netbeans software, the game was created and named as XML-KIDS. This game was described in UML, including its class and sequence diagrams. Finally, conclusions were presented and some ideas for future works were proposed.

Keywords: Ludic learning. XML. Serious games. Education

LISTA DE ILUSTRAÇÕES

Figura 1. Uma em cada 5 crianças utiliza dispositivos móveis todos os dias. Fonte: BLOG SONY XPERIA (2013) 14
Figura 2. Arcabouço Conceitual do Trabalho de Conclusão de Curso (TCC). Imagem adaptada pela autora. Fonte: <http://comps.canstockphoto.com.br/can-stock-photo_csp6122223.jpg> 18
Figura 3. Os objetos transformam-se em signos para a criança, (ELKONIN, 1998). Desenho de Calvin e Haroldo. Fonte: Bill Watterson. Disponível em: <https://torrenteliteraria.files.wordpress.com/2014/04/calvin-e-haroldo-capa.jpg> 21
Figura 4. Alguns padrões XML. Fonte: Ensino didático da linguagem XML. FURGERI (2001). 26
Figura 5. Vantagens da XML sobre a HTML. Fonte: FURGERI, 2001. 27
Figura 6. O sistema XML KIDS validando um documento XML. Fonte: (FURGERI, 2001, pág. 90) 31
Figura 7. Exemplo de DTD interno. Fonte: FURGERI (2001) págs. 98 e 99. 33
Figura 8. Os tipos de valores que um atributo pode assumir. Fonte: FURGERI (2001) pág. 121 35
Figura 9. Uso do atributo “pequena” no XML, com declaração do atributo como default no DTD. Fonte: FURGERI (2001) pág.122 35
Figura 10. Exemplo de XML com declaração de atributos no DTD. Fonte FURGERI (2001) pág. 125. 36
Figura 11. Exemplo de atributos no XML. Fonte: FURGERI (2001) pág. 127 36
Figura 12. O DTD valida o xml da página anterior. Fonte FURGERI (2001) pág. 128 37
Figura 13. Screenshot do jogo Touch Develop. Fonte:https://www.touchdevelop.com/ 39
Figura 14. Screenshot do jogo Touch Develop - Turtle. Fonte:https://www.touchdevelop.com/. Acesso em 03/maio/2017 39
Figura 15. Software Alice, para aprendizado de programação orientada a objetos e criação de jogos. Fonte: <https://upload.wikimedia.org/wikipedia/commons/thumb/5/57/Alice-2-screenshot.jpg/1200px-Alice-2-screenshot.jpg>. Acesso em: 28/09/2017 40
Figura 16. Curso de HTML Básico. W3Schools. Fonte: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro 40
Figura 17. Curso de HTML Básico. W3Schools. Fonte: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_headings 41
Figura 18. Objetos utilizados no MVC e suas interações. Fonte: <http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308> 42
Figura 19. Estrutura do Modelo proposto 46
Figura 20. Exemplo de Ambiente Casa. Fonte: https://pt.depositphotos.com/ 48
Figura 21. Tela Perfil do Usuário e Ambiente, com o código XML na parte inferior 51
Figura 22. Diagrama do Caso de uso Definir Perfil 52
Figura 23. Diagrama de Classes do Caso de Uso Definir Perfil do Usuário e Ambiente em MVC 53
Figura 24. Diagrama de sequência do Caso de Uso Definir perfil do usuário e ambiente 54
Figura 25. Tela do caso de uso Jogar. Na parte central da tela serão inseridos os objetos que se encontram na parte superior. Na parte inferior da tela é

exibido o código XML 55
Figura 26. Diagrama do Caso de Uso Jogar 56
Figura 27, Diagrama de classes do CDU Jogar em MVC 57
Figura 28. Diagrama de Sequência do CDU Jogar 58
Figura 29. Tela Caso de Uso Personalizar objetos 59
Figura 30. Caso de Uso Personalizar objetos 60
Figura 31. Diagrama de Classes do caso de uso Personalizar objetos 61
Figura 32. Diagrama de sequências do caso de uso Personalizar objetos 62
Figura 33. Diagrama de classes de entidades do modelo proposto 63

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Program Interface*
GUI	*Graphical User Interface* - Interface Gráfica de Usuário
HTML	*Hypertext Markup Language* - Linguagem de Marcação de Hipertexto
IFSC	Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina
JPA	*Java Persistence API*
RDF	*Resource Description Framework* - Estrutura de descrição de recursos
TCC	Trabalho de conclusão de curso
UML	*Unified Modeling Language* - Linguagem de modelagem unificada
W3C Brasil	*World Wide Web Consortium* - Escritório Brasileiro do Consórcio World Wide Web
XML	*eXtensible Markup Language* - Linguagem de marcação extensível

Sumário

1. INTRODUÇÃO 13
1.1. JUSTIFICATIVA 13
1.2. DEFINIÇÃO DO PROBLEMA 16
1.3. OBJETIVOS 16
1.3.1. Objetivo Geral 16
1.3.2. Objetivos Específicos 16
1.4. ESTRUTURA 17
1.5. METODOLOGIA 17
2. OS QUATRO PILARES DO REFERENCIAL TEÓRICO 18
O referencial teórico deste trabalho está fundamentado em quatro pilares: 18
2.1. O PROCESSO DE APRENDIZAGEM 19
2.2. A TECNOLOGIA DAS LINGUAGENS DE MARCAÇÃO E A WEB-SEMÂNTICA 22
2.2.1. *HyperText Markup Language* (HTML) 22
2.2.2. eXtensible Markup Language (XML). 22
2.3. ENSINO DE LINGUAGENS ATRAVÉS DE JOGOS 38
2.3.1. Serious Games 38
2.3.2 Exemplos de jogos para aprendizagem de linguagens 38
2.4. METODOLOGIA PARA ANÁLISE DO MODELO PROPOSTO. 42
2.4.1 Padrão de arquitetura MVC. 42
2.4.2. A linguagem UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) e seus diagramas. 43
3. DESENVOLVIMENTO DO TRABALHO – ESPECIFICAÇÃO DO MODELO 47
3.1. DESCRIÇÃO TEXTUAL DO MODELO BASEADO NO PROBLEMA (ITEM 1.2). 47
3.2. REQUISITOS FUNCIONAIS 49
3.3. CASOS DE USO 50
3.3.1. Caso de Uso Definir Perfil e Ambiente 50
3.3.2. Caso de Uso Jogar 54
3.3.3. Caso de Uso Personalizar objetos 58
3.3.4. Diagrama de classes de entidades de banco de dados. 62
4. CONCLUSÕES 64
5. RECOMENDAÇÃO PARA TRABALHOS FUTUROS 65
REFERÊNCIAS 66

1. INTRODUÇÃO

Este Trabalho de Conclusão de Curso (TCC) - requisito para a conclusão do curso Superior em Gestão da Tecnologia da Informação - propõe um modelo para aprendizagem da linguagem de marcação *eXtensible Markup Language* (XML). O modelo está baseado em jogos de computador, do tipo *serious games*, com ênfase em aplicativos como os presentes em *tablets* e visando atividades lúdicas e criativas que incentivem a aprendizagem da linguagem de marcação XML. Este modelo propõe a aprendizagem de forma lúdica, utilizando um jogo digital para computador em dispositivos móveis tipo *tablets* aproveitando a intuitividade das crianças com jogos e com os dispositivos com '*touchscreen*.

Este TCC está organizado em quatro partes. A primeira está composta da Introdução, Justificativa e Objetivos, onde se estabelece o escopo do trabalho. A segunda parte é identificada pelo título: Os Quatros Pilares do Referencial Teórico, onde são descritas as tecnologias que fundamentam o modelo proposto. A terceira parte – representando o desenvolvimento do trabalho – é composta pela especificação do modelo proposto. A quarta e última parte apresenta as conclusões, recomendações para trabalhos futuros e referências.

A metodologia científica empregada neste trabalho é a de uma pesquisa de finalidade aplicada e de abordagem qualitativa.

1.1. JUSTIFICATIVA

As pesquisas (figura 1) apontam que muitas crianças já usam dispositivos móveis todos os dias, isto é, desde muito cedo têm habilidades para operar computadores com *touchscreen*.

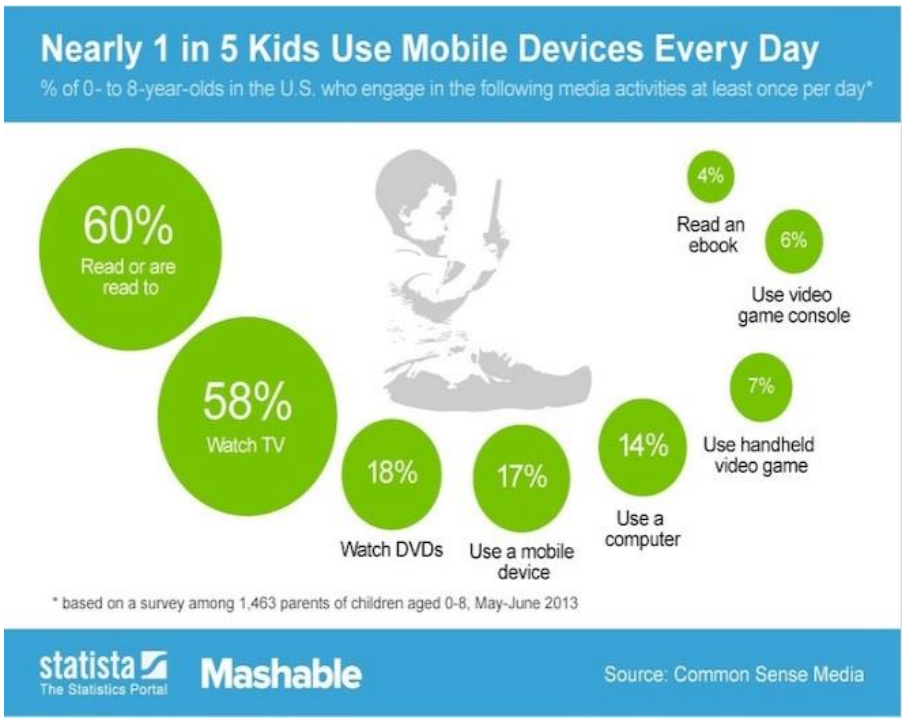


Figura 1. Uma em cada 5 crianças utiliza dispositivos móveis todos os dias. Fonte: BLOG SONY XPERIA (2013)

Naturalmente as crianças têm atração por jogos e animações (SANTOS, et. al. 2015) e, mesmo antes de aprenderem a falar, ler ou escrever, já conseguem usar esses equipamentos. Essa capacidade que a criança tem de usar esses recursos de forma intuitiva, leva a crer que esse pode ser um ponto de partida para começar a entender e usar uma linguagem de programação menos complexa, como as linguagens de marcação. Exemplos dessa abordagem podem ser vistos em vários artigos na Internet, como na recente entrevista sobre ensino de programação, com o diretor geral do Code Club Brasil, Felipe Fernandes, e o professor de Ciência da Computação da UFABC, Francisco Isidro Massetto (VEJA.COM, 2017). Os entrevistados destacam que “as ferramentas de programação de computadores facilitam a contextualização de outros temas e disciplinas”, portanto ajudam no aprendizado da escola formal (matemática, história, geografia, etc) e destacam também que aprender programação ajuda a sistematizar os problemas cotidianos e a encontrar soluções.

Qual a diferença de se ensinar uma linguagem de programação convencional como C, Java, Pascal etc. e uma linguagem de marcação como

XML? Talvez a forma simples e intuitiva das linguagens de marcação possa ter uma inserção mais fácil e precoce para a criança.

Além disso, espera-se que, em um futuro próximo, qualquer informação inserida no computador seja interpretada sem necessidade de mediação humana (web-semântica/ontologias) - outro viés importante para o ensino de linguagens de marcação como XML. Segundo as definições no site do *World Wide Web Consortium* (W3C), "Vocabulários e ontologias são importantes ferramentas e valiosos instrumentos para organizar os dados de um domínio" (W3C, 2017).

- A Web semântica é um desafio para a construção do conhecimento. Como tal, requer todo um arcabouço de definição para sustentá-la, como por exemplo, XML (*eXtensible Markup Language*), RDF *(Resource Description Framework*) , e ontologias (W3C, 2017).
- Essa arquitetura de expressão do conhecimento agrega significados e contextos que permitem a independência de interpretações externas. Isso fará com que a informação seja melhor explicada. (W3C, 2017)
- A linguagem de marcação é essencial na adoção da tecnologia de Web Semântica, que é a tendência para o futuro, pois segundo o W3C,

> Os padrões XML estão onipresentes na computação empresarial, e fazem parte da base da Web. Por serem altamente interoperáveis e acessíveis, esses padrões vêm sendo adotados pelas pessoas em uma ampla variedade de aplicações (W3C, 2017).

A união da habilidade das crianças ao usar a *Graphical User Interface* (GUI) *touchscreen* com a tendência às tecnologias auto-explicadas sugere a possibilidade de iniciar um processo de construção de linguagem homem/computador de forma bem precoce, e de forma lúdica.

Portanto, aliar esses conceitos de aprendizagem intuitiva, disponibilidade de jogos e ferramentas para ensino de linguagens e web-semântica, numa experiência de ensino de linguagem de marcação, baseado em *tablets* e com a semântica de jogos para crianças, deve proporcionar mais oportunidades para o desenvolvimento pessoal e de novas tecnologias.

Desta forma, propõe-se aqui um exercício de criação de um modelo para contribuir com essa expectativa.

1.2. DEFINIÇÃO DO PROBLEMA

Este trabalho pretende contribuir com a discussão a respeito do aprendizado de linguagens de marcação. Ou seja, como estimular as crianças ao aprendizado da linguagem de marcação XML, de forma lúdica, baseado em jogos e na predisposição que as crianças têm para o uso de dispositivos móveis como tablets e smartphones?

1.3. OBJETIVOS

Para o desenvolvimento deste trabalho foram definidos os seguintes objetivos, geral e específicos.

1.3.1. Objetivo Geral

O objetivo geral deste trabalho é propor um modelo de aprendizagem lúdica de XML baseado em jogos de computador.

1.3.2. Objetivos Específicos

- Ampliar os conhecimentos e práticas obtidas nas Unidades de ensino do Curso Superior em Gestão de Tecnologia da Informação;
- Pesquisar métodos e técnicas para ensinar linguagens de computador para crianças;
- Explorar os conceitos relacionados às linguagens de marcação;
- Usar uma metodologia para a modelagem do sistema;
- Auxiliar no entendimento da linguagem de marcação XML de forma lúdica e em contextos de aprendizagem específicos.

1.4. ESTRUTURA

Este documento está estruturado em cinco capítulos:

1. Introdução, justificativa, definição do problema e objetivos; 2. Os Quatros Pilares do Referencial Teórico; 3. Especificação do modelo; 4. Conclusões; 5. Recomendações para trabalhos futuros.

Cada capítulo constitui uma parte fundamental do estudo feito para a elaboração deste trabalho de conclusão de curso.

1.5. METODOLOGIA

A metodologia científica empregada é a de uma pesquisa de finalidade aplicada, bibliográfica quanto aos procedimentos e de abordagem qualitativa, visando modelar uma solução - modelagem na linguagem UML - para facilitar a aprendizagem da linguagem XML. Esta pesquisa é Qualitativa porque o critério para identificação dos resultados não é numérico, mas valorativo; e é Aplicada porque é voltada à solução de um problema específico. Como pesquisa de cunho acadêmico não poderia deixar de ter um levantamento bibliográfico.

Após a identificação e delimitação do problema proposto para a pesquisa, foi realizada a pesquisa bibliográfica sobre os temas relacionados aos processos de aprendizagem, tecnologias que envolvem as linguagens de marcação (HTML e XML) e modelagem de sistemas. Na sequência definiu-se como o modelo seria implementado, partindo-se então para a análise e implementação dos protótipos. A base de implementação do modelo são os jogos estudados que visam a aprendizagem de linguagens, bem como das teorias de aprendizagem, do estado da arte nas implementações da linguagem XML e de uma metodologia que fosse capaz de modelar a aplicação para futuras implementações.

O modelo é proposto seguindo os seguintes passos: definição dos Casos de Uso e montagem dos diagramas em UML, usando como ferramenta principal os diagramas da linguagem UML e uma metodologia desenvolvida: descrição textual do problema, identificação dos requisitos funcionais, casos de uso, modelagem dos casos de uso e diagrama de entidades.

A Figura 2 representa o arcabouço conceitual do Trabalho de Conclusão de Curso (TCC).

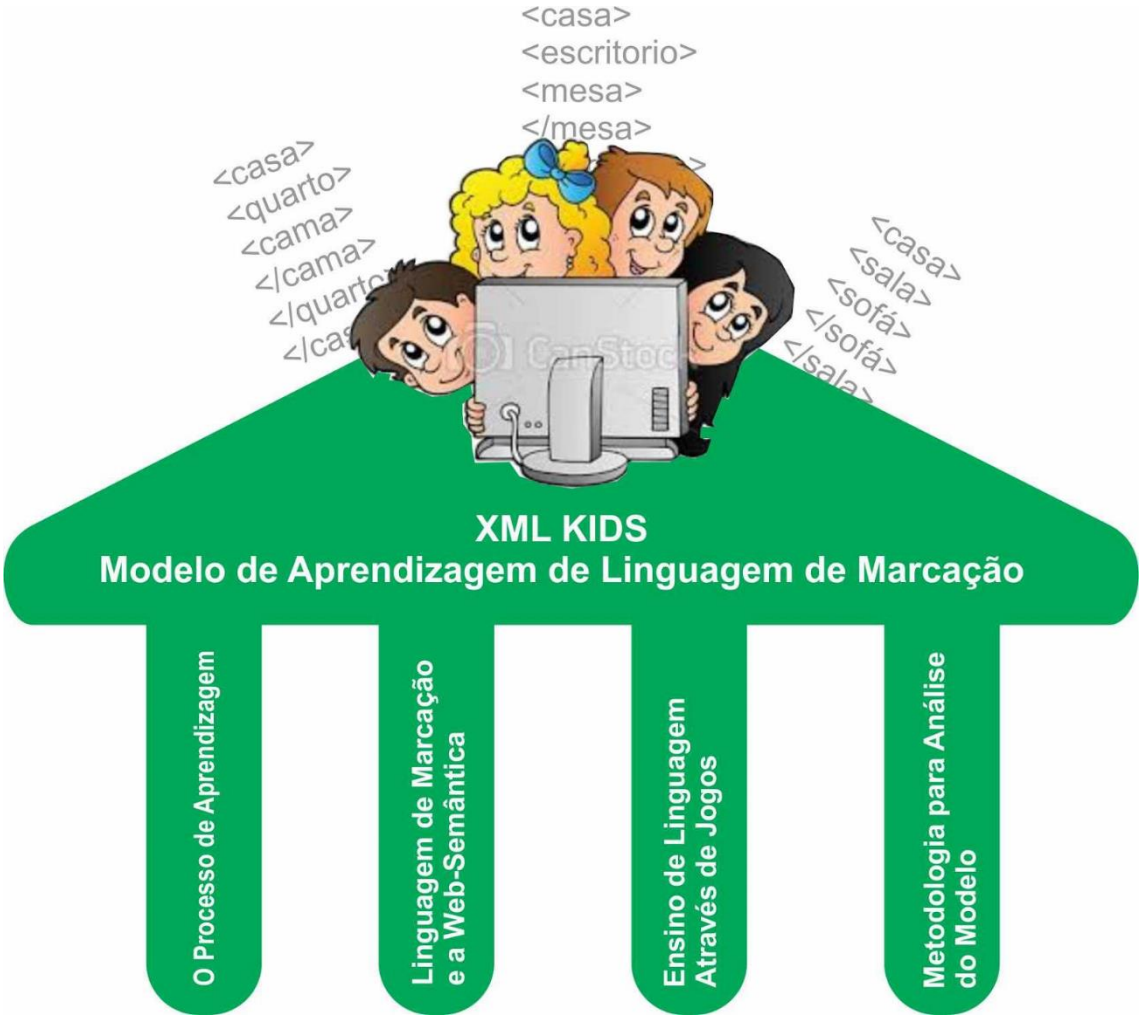


Figura 2. Arcabouço Conceitual do Trabalho de Conclusão de Curso (TCC). Imagem adaptada pela autora. Fonte: <http://comps.canstockphoto.com.br/can-stock-photo_csp6122223.jpg>

2. OS QUATRO PILARES DO REFERENCIAL TEÓRICO

O referencial teórico deste trabalho está fundamentado em quatro pilares:

- O processo de aprendizagem
- A tecnologia das linguagens de marcação e a Web semântica
- Ensino de linguagens através de jogos
- Metodologia para análise do modelo

2.1. O PROCESSO DE APRENDIZAGEM

Para Jean Piaget – 1896/1980 – (apud CHALONBLANC, 2001), o processo de desenvolvimento intelectual ocorre em 3 etapas: 1) **assimilação**: quando os elementos do meio externo são incorporados à estrutura mental existente; 2) **acomodação**: a assimilação provoca um processo de mudanças na estrutura; 3) **adaptação**: é o equilíbrio entre a assimilação da experiência nas estruturas dedutivas e a acomodação dessas estruturas aos dados da experiência.

De acordo com Piaget (apud CHALONBLANC, 2001) “O sujeito é um organismo que possui estruturas e, ao receber os estímulos do meio, produz uma resposta em função dessas estruturas”. Como tal, a evolução é concebida como um processo no qual o sujeito realiza a reconstrução do conhecimento, envolvendo a articulação de dois aspectos: 1) o **desenvolvimento intelectual** do sujeito, envolvendo a existência ou não de esquemas de assimilação adequados ao nível de instrução ministrada, ou de adaptação a essa instrução em seu nível de desenvolvimento intelectual; 2) a **tomada de consciência**, ou seja, a passagem da assimilação para a elaboração de conceitos, permitindo que o sujeito realize a construção dos conceitos abordados na instrução. (MODESTO e SCAVACINI, 2013).

A brincadeira é um elemento de tomada de consciência. Para pesquisadores como Negrine (1994), Friedmann (1996), Baptista da Silva (2003), Biscoli (2005) e o próprio Vygotsky (1991) não há diferença semântica entre as palavras “jogo” e “brincadeira”. Eles utilizam ambas as palavras para designar o mesmo comportamento, **a atividade lúdica**. Negrine (1994), explica que o termo jogo vem do latim *iocus* e significa diversão, brincadeira. Para Brougère e Wajskop (1997) existem algumas diferenças entre brincadeira e jogo. Esses autores afirmam que a brincadeira é simbólica e tem a característica de ser livre, e ter um fim em si mesma, enquanto o jogo é funcional e tem um objetivo final, a vitória. Por ter um objetivo final, tem regras pré-estabelecidas. As regras dos jogos, que já chegam prontas para as crianças, estão relacionadas intimamente com as regras sociais, morais e culturais existentes.

Mesmo recebendo as regras prontas, as crianças têm a liberdade e a flexibilidade de aceitar, modificar ou simplesmente ignorá-las, tendo ao mesmo

tempo influência do contexto e dos parceiros de jogo. No jogo, o objetivo final da criança é a vitória sobre o oponente, mas, mesmo que a criança não vença, o prazer experimentado ao jogar pode fazer com que a criança deseje jogar novamente (BROUGÈRE, 1998). Portanto, o jogo também tem um fim em si mesmo. A vitória final já não é tão importante, mas sim o processo. Considerando estas características, de flexibilidade, de prazer e de fim em si mesmo, o brincar e o jogar são indistintos (CORDAZZO e VIEIRA, 2007).

Vygotsky (1991) afirma que a brincadeira é uma situação imaginária criada pela criança e onde ela pode, no mundo da fantasia, satisfazer desejos até então impossíveis para a sua realidade. Portanto, o brincar "é imaginação em ação" (FRIEDMANN, 1996).

Winnicott (1975) explica que "É no brincar, e talvez apenas no brincar, que a criança ou o adulto fruem na sua liberdade de criação... É no brincar, e somente no brincar, que o indivíduo, criança ou adulto, pode ser criativo e utilizar sua personalidade integral: e é somente sendo criativo que o indivíduo descobre o eu (self)".

Segundo Vygotsky (1991), os jogos oferecem ambientes desafiadores, capazes de "estimular o intelecto" permitindo conquistar níveis mais avançados de raciocínio. Quando situações de jogos são propostas na sala de aula, ocorrem momentos de afetividade entre a criança e o aprender, e a aprendizagem se torna mais significativa e prazerosa.

Ainda sob a ótica de Vygotsky (1991), jogos e brincadeiras inseridos no processo de ensino, são excelentes para a aprendizagem, além de ajudar no desenvolvimento físico, motor, intelectual, psicológico, e aprimorar as habilidades sociais, a interação e o raciocínio das crianças.

Para Vygotsky (1991), a brincadeira nasce da necessidade de um desejo frustrado pela realidade. Elkonin (1998) amplia essa sentença quando afirma que os objetos, ao terem seus significados substituídos, transformam-se em signos para a criança (Figura 3). Desta forma, "a criança não faz distinção entre o brinquedo e o que ele significa, mas a utilidade que terá nas representações que serão feitas com ele" (BISCOLI, 2005; p. 29, apud CORDAZZO e VIEIRA, 2007)

Figura 3. Os objetos transformam-se em signos para a criança, (ELKONIN, 1998). Desenho de Calvin e Haroldo. Fonte: Bill Watterson. Disponível em: <https://torrenteliteraria.files.wordpress.com/2014/04/calvin-e-haroldo-capa.jpg>

Como essas definições se relacionam com uma brincadeira numa atividade em um computador, onde a criança possa, a partir de um objeto, criar um significado pessoal daquele objeto? Nesse sentido, muitos jogos em computador disponibilizam personagens e atributos que podem ser utilizados de diferentes formas.

Uma pesquisa de 2013 da Common Sense Media, realizada nos Estados Unidos e publicada pelo Blog Sony Xperia, indica que 38% das crianças com até 2 anos de idade já utilizaram aparelhos mobile, como *smartphones* e *tablets*. A medida que as crianças vão crescendo, a porcentagem aumenta, e vai para 72% entre crianças de ambos os sexos de até 8 anos de idade. Nessa faixa etária pode se dizer que o uso da tecnologia já é um hábito. Nessa pesquisa, 17% das crianças de 8 anos usa pelo menos um aparelho *mobile* todos os dias.

Se a maior parte da experiência cultural das crianças vem do repertório das mídias, sobretudo a partir do uso dos *tablets*, e, considerando que na nossa cultura os dispositivos móveis têm uma enorme inserção na vida das pessoas, é necessário considerar essa questão na educação infantil. Como Silverstone (2003) afirma: a alfabetização em mídias se torna “fundamental para a construção de identidades, o senso de nós mesmos no mundo e nossa capacidade de agir dentro dele” (SILVERSTONE, 2003, p. 147).

Segundo Fantin (2008, 2012), para saber como as crianças se relacionam com as novas coisas do mundo, devemos pensar nelas “interagindo com a cultura a partir das transformações da sociedade contemporânea”. Ou seja, a

noção de alfabetismo é “a necessidade de circular por outros tipos de representações da realidade” que, além da escrita, envolvem as representações visuais, musicais, corporais, digitais, informáticas e outras. Essas dimensões devem ser trabalhadas de maneira transdisciplinar, analisando as “diferentes representações do próprio conhecimento” incluindo as formas culturais, importantes na sociedade contemporânea. Portanto, deve ser identificada essa noção de alfabetismo como alfabetização nas múltiplas linguagens, com ênfase na plena circulação e interação entre as linguagens. Com esse conceito, pode se falar em “alfabetização científica, alfabetização literária, artística, estética e cultural” incluindo a alfabetização tecnológica (FANTIN, 2008, 2012).

2.2. A TECNOLOGIA DAS LINGUAGENS DE MARCAÇÃO E A WEB-SEMÂNTICA

Para fundamentar a conveniência do ensino de XML, alguns conceitos básicos sobre as linguagens de marcação e a Web-Semântica são explicados a seguir.

2.2.1. *HyperText Markup Language* (HTML)

HTML é uma linguagem de marcação utilizada na construção de páginas na Web e seu objetivo é “realizar apenas a formatação do texto” (FURGERI, 2001). Os documentos em formato HTML podem ser interpretados pelos navegadores (*browsers*) por meio de marcadores (*tags*), que são comandos de formatação da linguagem pré-definidos por seus criadores com o objetivo de fazer a marcação de cada parte do documento HTML. (FURGERI, 2001).

Em virtude do ineficiente mecanismo de busca de informações e de outros procedimentos não atendidos pela HTML, surgiu a XML, uma linguagem criada com a finalidade de fornecer significado às informações armazenadas nos documentos (FURGERI, 2001).

2.2.2. eXtensible Markup Language (XML).

De acordo com a definição apresentada no site do W3C (2017), organização oficial que desenvolve padrões para criação de conteúdos web, a linguagem de

marcação extensível XML é um formato simples, baseado em texto, para representação de informações estruturadas, como documentos, dados, configuração, livros, etc. Esta linguagem foi derivada de um formato padrão anterior, o SGML (*Standard Generalized Markup Language*) para ser mais compatível para o uso da Web. O XML é um dos formatos mais amplamente utilizados hoje em dia para compartilhar informações estruturadas: entre programas, entre pessoas, entre computadores e pessoas, tanto de forma local como através das redes (W3C, 2017).

Conforme explicado por Furgeri (2001), na XML não existe nenhuma *tag* predefinida; o usuário cria suas próprias *tags* criando assim a sua linguagem de marcação. XML na verdade é uma metalinguagem, ou seja uma linguagem usada para criar outra linguagem.

Por gerenciar as informações de modo eficiente e inteligente, a linguagem XML produz uma rede mais organizada. Por esse motivo ela tem recebido um grande destaque na área de informática (FURGERI, 2001).

2.2.2.1. A importância da XML

“Enquanto a HTML apenas trata de especificar a formatação de uma palavra ou um trecho de texto, a XML trata de fornecer seu significado. Esse é o aspecto mais importante da XML” (FURGERI, 2001).

A XML permite descrever o documento de forma inteligente, tornando o conteúdo mais compreensível para os seres humanos bem como para os computadores. “Enquanto a HTML indica como algo deve ser exibido, a XML indica o que a informação significa” (FURGERI, 2001).

A XML é um aperfeiçoamento da HTML, e possibilita o desenvolvimento de automatização de diversas atividades comerciais por meio da Web. A semelhança com a HTML é que não é uma linguagem de programação, e também é um padrão aberto. A XML permite uma padronização na descrição de informações. Com a estrutura do documento padronizada, é possível, mediante a utilização de linguagens de programação, interpretar e manipular o conteúdo do documento (FURGERI, 2001).

Furgeri (2001) explica que um documento XML é composto basicamente de 3 (três) elementos:

1. Conteúdo dos dados: são as informações armazenadas entre as tags.
2. Estrutura: é a organização dos elementos dentro do documento. Este pode ter vários tipos de formato (um contrato, uma receita, um orçamento, etc.), dependendo das necessidades de marcação da informação.
3. Apresentação: é a forma de apresentar o conteúdo de um documento XML para o leitor do documento. Um mesmo documento pode ser visualizado de formas diferentes.

A XML oferece uma forma padrão para descrever, capturar, processar e publicar informações agrupadas em um documento XML. Ela possibilita tornar a Internet mais inteligente, transformando o modo de armazenar e recuperar informações, e promovendo redução de custos operacionais (FURGERI, 2001).

2.2.2.2. Estilo da XML

A seguir, um exemplo comparando um documento HTML com um documento XML, com o mesmo conteúdo de texto:

Em HTML:

```
<HTML>
<BODY>
Casa grande, com mesa de jantar, cadeiras de jantar, sofá, louças.
</BODY>
</HTML>
```

Em XML:

```
<?xml version="1.0"?>
<CASA tipo="grande">
    <MESA tipo="Jantar">
      <CADEIRA>de jantar</CADEIRA>
      <CADEIRA>de jantar</CADEIRA>
      <CADEIRA>de jantar</CADEIRA>
      <CADEIRA>de jantar</CADEIRA>
     '<LOUÇA>prato</LOUÇA>
      <LOUÇA>xícara</LOUÇA>
    </MESA>
    <SOFA>2 lugares</SOFA>
</CASA>
```

No exemplo anterior, é possível ver claramente que a XML torna o documento mais inteligente, demonstrando qual o significado das informações, que estão agrupadas no documento. Furgeri (2001) explica que no documento HTML, um

software não pode reconhecer o significado dos dados inseridos, pois estão misturados em um texto corrido. Já no documento XML, as informações estão separadas e organizadas, de modo que tanto o leitor humano, como o software utilizando linguagens de programação, poderão interpretar seu significado, seja de forma integral ou em trechos separados (FURGERI, 2001).

Nesse exemplo, verifica-se que as *tags* foram criadas de acordo com a necessidade do desenvolvedor do documento, sem a limitação imposta pela HTML, onde só podem ser utilizadas as *tags* definidas pela linguagem.

Também poderiam ser utilizadas outras *tags* para definir melhor os móveis da casa:

```
<?xml version="1.0"?>
<CASA tipo="grande">
  <MESA tipo="Jantar">
    <CADEIRA tipo="jantar">
      <MATERIAL>Madeira</MATERIAL>
      <QUANTIDADE>4</QUANTIDADE>
    </CADEIRA>
    <LOUÇA>
        <PRATO tipo="raso"></PRATO>
        <XICARA tipo="cha"></XICARA>
    </LOUÇA>
  </MESA>
  <SOFA>2 lugares</SOFA>
</CASA>
```

Neste exemplo, foram criadas as *tags* <MATERIAL> e <QUANTIDADE> para detalhar as características da cadeira e foram agrupadas as *tags* <PRATO> e <XICARA> dentro da *tag* <LOUÇA>.

O mais importante é que diversos padrões para marcação de documentos podem ser construídos baseados na estrutura que a XML oferece. (FURGERI, 2001).

Como Furgeri (2001) explica, já existem vários padrões no mercado para estabelecer uma forma única entre os usuários. Alguns desses padrões estão detalhados na Figura 4, a seguir:

MathML	Mathematical Markup Language, utilizada para descrição de fórmulas matemáticas
MML	Medical Markup Language, para troca de dados entre instituições envolvidas com clínica médica
MML	Music Markup Language, para marcar diferentes tipos de eventos de música
GXL	Graph Exchange Language, para padronizar a troca de elementos gráficos
HCML	Health Care Markup Language, utilizada na área médica para planos de saúde
AML	Astronomical Markup Language, para padronização de elementos na área de astronomia
CML	Chemical Markup Language, utilizada na marcação de informações na área química
ComicsML	Comics Markup Language, para definir padrões para a criação de histórias em quadrinhos

Figura 4. Alguns padrões XML. Fonte: Ensino didático da linguagem XML. FURGERI (2001).

E continuam surgindo padrões em várias áreas para representar com precisão o significado do conteúdo dos documentos XML, permitindo que “várias pessoas ou programas saibam exatamente o significado de cada trecho do documento” (FURGERI, 2001), facilitando a troca de informações, de maneira automática pela Internet.

2.2.2.3. Vantagens da XML sobre a HTML

Na Figura 5 a seguir são exibidas algumas das principais vantagens da XML sobre a HTML.

HTML	XML
Não permite definição de tags. Trabalha apenas com tags pré-definidos	Possibilita a criação de tags conforme as necessidades do usuário e o surgimento de linguagens baseadas em XML.

A pesquisa é realizada de forma “burra”, as palavras não possuem significado.	O conteúdo do documento pode ser pesquisado de forma inteligente. Exemplo: <CD> <CANTOR>Nome do cantor</CANTOR> <TITULO>Nome do título</TITULO> <GRAVADORA>Nome da gravadora</GRAVADORA> </CD> Possibilitando a criação de um sistema de busca mais especializado, com informações mais úteis e precisas.
Os documentos são visualizados sempre da mesma maneira (com a mesma formatação e ordenação)	Um mesmo documento pode ser visualizado de diferentes formas, utilizando folhas de estilo, ordenando e filtrando informações segundo certos critérios

Figura 5. Vantagens da XML sobre a HTML. Fonte: FURGERI, 2001.

Furgeri (2001) explica outras vantagens da XML:

- A estrutura do documento XML permite que ferramentas de bancos de dados possam consultar e processar o conteúdo.
- Ferramentas desenvolvidas para manipular XML facilitam a criação de documentos, já que o desenvolvedor se dedica apenas ao conteúdo, pois a estrutura é controlada pela ferramenta.
- A estrutura XML possibilita a automatização dos processos, por meio de *softwares* que reconhecem as *tags* do documento XML, integrando o site na Internet com o software interno da empresa, reduzindo custos e tornando os processos mais eficientes.
- A XML tende a ser um padrão permanente, independente das mudanças na área de informática, já que o conteúdo do documento é puramente texto, e poderá ser lido e atualizado por qualquer ferramenta futura.
- Aplicações padronizadas para XML permitem que diferentes aplicativos trabalhem em conjunto, com maior interoperabilidade na troca de informações.
- Um documento em XML é mais eficiente, pois permite automatizar grande parte das tarefas que, com a linguagem HTML, eram realizadas manualmente.
- Os browsers podem processar grande parte das informações mediante o uso de linguagens de programação. O conteúdo do documento pode ser manipulado

e reorganizado. Cálculos podem ser realizados para gerar novos conteúdos e novos documentos.

- Os recursos fornecidos pela XML podem ser utilizados para criar uma rede de conhecimento (*Knowledge Web*), interligando documentos com informações complementares, mesmo estando em lugares diferentes ou pertencendo a diferentes empresas (FURGERI, 2001).

2.2.2.4. Onde utilizar a XML

Segundo FURGERI (2001), as principais aplicações da XML envolvem a troca eletrônica de documentos para automatizar os processos de negócio entre empresas (B2B - *Business to Business*), ou processos de negócios entre empresas e consumidores (B2C - *Business to Consumer*). Nas duas modalidades podem surgir diversas ramificações para o uso de aplicações que pesquisam e manipulam documentos em XML, algumas possibilidades serão explicadas a seguir.

Usos da XML

a) Para melhorar a busca de informações

Para agilizar a busca na Web, é necessário padronizar os mecanismos de busca. As páginas da Web, em HTML, incluem informações sobre formatação, mas a HTML não fornece uma maneira de descrever os conteúdos do texto, portanto o significado se perde por não ter uma forma de demarcar isto. É necessário criar páginas dinâmicas, para apresentar o conteúdo conforme a necessidade do usuário. Por exemplo, ordenar uma lista por nome, cor, preço, ou fabricante.

Usando a XML, é possível tornar a busca mais inteligente, utilizando tags para especificar os locais da página onde deve ser efetuada a pesquisa, e evitando páginas irrelevantes (FURGERI, 2001).

b) Para Automação da Web

Devido a que a XML permite definição de marcadores especiais, e portanto é possível definir o significado dos dados, o documento adquire uma estrutura bem definida, e com isso, ferramentas de software desenvolvidas em qualquer

linguagem de programação podem manipular seu conteúdo, aprimorando a pesquisa. A Internet fica mais eficiente e ágil em suas atualizações (FURGERI, 2001).

c) Para troca de dados na Web

Por existirem diversos tipos de software, sobre diferentes plataformas, os sistemas de computador e bancos de dados têm dados em formatos incompatíveis. A XML permite que computadores diferentes troquem dados pela Internet, pois o conteúdo dos documentos XML é apenas texto. Para isso basta converter o formato dos dados armazenados para o formato texto (FURGERI, 2001).

d) Em conjunto com documentos HTML

Documentos HTML exibem dados de todo tipo. Com a XML os dados ficam armazenados em “ilhas” (arquivos separados) interligadas ao documento HTML. O arquivo XML pode ser alterado a qualquer momento e essas mudanças serão refletidas no documento HTML na tela, sem necessidade de atualização (FURGERI, 2001).

e) Para o desenvolvimento de catálogos

Um catálogo é parte fundamental de um site, pois exibe produtos, serviços, fotos, preços, etc. Os catálogos escritos em linguagem XML permitem diversos métodos de acesso e formas de apresentação pois diversas ferramentas podem analisar seu conteúdo e exibir as informações em diferentes formatos. Essas informações podem ser classificadas de várias maneiras (por nome, por preço, por tamanho, etc) (FURGERI, 2001).

f) Para automação da cadeia de suprimentos

A automação da cadeia de suprimentos pode ser descrita como um grupo de empresas parceiras trocando informações sobre seus produtos para realizarem suas negociações.

Os sistemas criados a partir da XML permitem a automação do processo de compras e maior flexibilidade e agilidade na escolha de fornecedores, facilitando os processos na área de compras (FURGERI, 2001).

g) Outras aplicações

A XML pode ser utilizada em qualquer aplicação em que seja necessária a troca de dados de forma padronizada, por exemplo, em transações bancárias pela Internet, envio eletrônico de dados, publicações científicas, aplicações comerciais ou em qualquer site da Web (FURGERI, 2001).

2.2.2.5. DTD. Declaração do tipo de documento

Conforme a definição de FURGERI (2001), o DTD (Document Type Definition), documento em formato texto, contém todas as regras estabelecidas para elaborar um documento XML, indicando todas as partes obrigatórias e opcionais dele.

Na prática, todo documento XML deve estar atrelado a um DTD, e estes dois documentos trabalham em conjunto em uma página da Internet (FURGERI, 2001).

A função do DTD é “definir todas as *tags* que um documento XML pode conter, determinando a ordem em que elas devem aparecer e se são obrigatórias ou opcionais” (FURGERI, 2001).

O DTD é responsável por armazenar também os atributos que as tags podem conter, as entidades utilizadas nos documentos, e outros elementos do documento XML (FURGERI, 2001).

Com o DTD, o software processador da XML pode validar um documento, verificando se este foi criado corretamente, segundo os padrões estabelecidos.

Quando se utiliza HTML em um página da Web, não é necessário o DTD porque suas tags são fixas e podem ser controladas internamente pelo navegador. Já a XML permite a criação de *tags*, portanto o DTD é necessário para que o navegador possa verificar a validade das *tags* utilizadas. O DTD permite que o navegador trabalhe como um “compilador” de documentos XML (FURGERI, 2001).

O sistema XML KIDS, modelo proposto neste trabalho, será o software processador da XML, e o DTD terá as tags do XML KIDS.

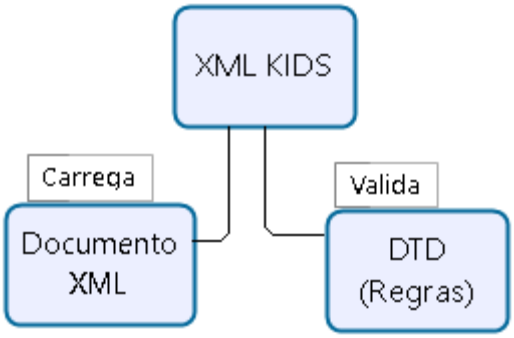


Figura 6. O sistema XML KIDS validando um documento XML. Fonte: (FURGERI, 2001, pág. 90)

Com o DTD, o sistema verifica todos os detalhes armazenados no documento XML e se encontrar um erro, este será informado na tela. Para realizar a validação, o documento deve ser bem formado, com todas as tags de abertura, cada uma delas com sua correspondente tag de encerramento. O DTD pode ser definido de maneira externa (em um arquivo separado) ou interna (dentro do documento XML) (FURGERI, 2001).

Utilizando o DTD se pretende estabelecer padrões de marcação para todas as áreas possíveis, permitindo que todas as empresas, do mesmo ramo de atividade ou de ramos diferentes, possam se comunicar de maneira unificada (FURGERI, 2001).

Ao construir o DTD, deve ser definido claramente que tipo de informação será armazenado no documento XML, e todos os outros elementos, tais como atributos, entidades, etc, podendo alterar o conteúdo do DTD e inserir mais informações quando for necessário (FURGERI, 2001).

Segundo FURGERI (2001), o DTD deve reconhecer os seguintes blocos de construção:

a. Elementos: blocos de construção utilizados em HTML bem como em XML. Alguns elementos em HTML são <BODY>, <TABLE>, etc. Em XML pode ser qualquer *tag* criada.

b. Tags: elementos marcadores com abertura e encerramento.

c. Atributos: inseridos nas tags iniciais de um elemento, fornecem informações adicionais sobre os elementos. Os atributos têm um nome e um conteúdo (por exemplo, <IMG **src=”foto.gif”**>

d. Entidades: variáveis designadas para conter textos ou documentos.

e. Notações: há duas formas de notação:

1. PCDATA que permite armazenamento de texto que será interpretado pelo analisador do documento; e
2. CDATA que armazena texto não interpretado pelo analisador.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<!-- Este é um DTD interno para livros -->
<!DOCTYPE LIVROS [
<!ELEMENT LIVROS ANY>
<!ELEMENT LIVRO
(TITULO,AUTOR_SOBRE,AUTOR_NOME,AREA,ANO,PAGINAS,PRECO,EDITORA)>
<!ELEMENT TITULO (#PCDATA)>
<!ELEMENT AUTOR_SOBRE (#PCDATA)>
<!ELEMENT AUTOR_NOME (#PCDATA)>
<!ELEMENT AREA (#PCDATA)>
<!ELEMENT ANO (#PCDATA)>
<!ELEMENT PAGINAS (#PCDATA)>
<!ELEMENT PRECO (#PCDATA)>
<!ELEMENT EDITORA (#PCDATA)>
]>
<LIVROS>
 <LIVRO>
      <TITULO>XML</TITULO>
      <AUTOR_SOBRE>Furgeri</AUTOR_SOBRE>
      <AUTOR_NOME>Sérgio</AUTOR_NOME>
      <AREA>Internet</AREA>
      <ANO>2001</ANO>
      <PAGINAS>300</PAGINAS>
      <PRECO>54,00</PRECO>
      <EDITORA>Érica</EDITORA>
 </LIVRO>

 <LIVRO>
      <TITULO>Business to Business</TITULO>
      <AUTOR_SOBRE>Furgeri</AUTOR_SOBRE>
      <AUTOR_NOME>Sérgio</AUTOR_NOME>
      <AREA>Informática e Negócios</AREA>
      <ANO>2001</ANO>
      <PAGINAS>242</PAGINAS>
      <PRECO>45,00</PRECO>
      <EDITORA>Érica</EDITORA>
 </LIVRO>
</LIVROS>
```

Figura 7. Exemplo de DTD interno. Fonte: FURGERI (2001) págs. 98 e 99.

2.2.2.6. Atributos

Furgeri (2001) explica que os atributos são identificadores especiais inseridos nos documentos XML para fornecer valores aos elementos inseridos nesses documentos.

Atributos também devem ser declarados no DTD para que o processador responsável pela interpretação do documento possa realizar sua validação.

Os atributos não são obrigatórios nos documentos XML, mas seu uso permite elaborar documentos mais especializados.

Atributos podem funcionar como rótulos identificadores. (FURGERI, 2001)

Eles são inseridos nas *tags*:

Por exemplo:

<UNIVERSIDADE nome=”Uni1”>

<UNIVERSIDADE nome=”Uni2”>

<UNIVERSIDADE nome=”Uni3”>

Cada atributo tem um identificador (nome) e um conteúdo (“Uni1”)

Nesse exemplo, o atributo “nome” foi utilizado para diferenciar documentos de diferentes instituições.

Formato para utilização de atributos

Os atributos são palavras adicionadas às tags e a outros elementos do documento XML; eles fornecem informações complementares, normalmente de tamanho pequeno.

Em HTML, um atributo pode ser usado para definir o alinhamento de um parágrafo, por exemplo: <P align=”left”>. Nos documentos XML, os atributos são utilizados também na tag de início, com o seguinte formato:

<tag [nome_do_atributo]=”[valor_do_atributo]”>.

Sempre que um atributo for utilizado em XML, ele deve ter um valor. (FURGERI, 2001)

Tipo	O valor do atributo pode ser
CDATA	Um texto qualquer, exceto os caracteres que possuem um significado especial para o interpretador: "<", ">" e "&"
(val1|val2|...)	Uma lista de valores
ID	Um identificador único
IDREF	Um identificador de outro elemento
IDREFS	Uma lista de identificadores de outros atributos
NMTOKEN	Um nome válido em XML
NMTOKENS	Uma lista de nomes válidos em XML
ENTITY	Uma entidade
ENTITYS	Uma lista de entidades
NOTATION	Uma notação

Figura 8. Os tipos de valores que um atributo pode assumir. Fonte: FURGERI (2001) pág. 121

Inserindo uma declaração no DTD, o atributo é opcional:

<!ATTLIST EMPRESA tipo CDATA “pequena”>

Esta declaração indica que a tag EMPRESA, utilizada no documento XML, pode conter um atributo chamado tipo. Usando a palavra reservada CDATA, o valor desse atributo pode ser qualquer string de caracteres (exceto os caracteres “&”, “<”, “>”).

No caso desta declaração o atributo é opcional e se não tiver nenhum valor no documento XML, será atribuído o valor default “pequena” (Figura 9).

```
<EMPRESA tipo = "pequena"> .... </EMPRESA>
<EMPRESA tipo = "média"> .... </EMPRESA>
<EMPRESA tipo = "grande"> .... </EMPRESA>
<EMPRESA> .... </EMPRESA>
```

Figura 9. Uso do atributo “pequena” no XML, com declaração do atributo como default no DTD. Fonte: FURGERI (2001) pág.122

A Figura 10 mostra um exemplo de XML com declaração de atributos no DTD.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<!DOCTYPE CADASTRO SYSTEM "ex0801.dtd">
<CADASTRO>
 <PRODUTO cod1 = "P01">Teclado</PRODUTO>
 <PRODUTO cod1 = "P02">Mouse</PRODUTO>
 <PRODUTO cod1 = "P03">Monitor</PRODUTO>
 <PRODUTO cod1 = "P04">Microprocessador</PRODUTO>
 <ACESSORIO cod2 = "P02">MousePad</ACESSORIO>
 <ACESSORIO cod2 = "P04">Cooler</ACESSORIO>
</CADASTRO>

<!—aqui começa o DTD externo com a lista de identificadores -->
<!ELEMENT CADASTRO ANY>
<!ELEMENT PRODUTO (#PCDATA)>
<!ELEMENT ACESSORIO (#PCDATA)>
<!ATTLIST PRODUTO cod1 ID #REQUIRED>
<!ATTLIST ACESSORIO cod2 IDREF #REQUIRED>
```

Figura 10. Exemplo de XML com declaração de atributos no DTD. Fonte FURGERI (2001) pág. 125.

A Figura 11 é um exemplo de definição de atributos no XML

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<!DOCTYPE AGENDA SYSTEM "ex0803.dtd">
<AGENDA mes="Maio">
 <DIA numero="D01" tipo="util" semana="quarta">
 <HORA valor="8.00">Reunião com a Diretoria</HORA>
 <HORA valor="10.00">Reunião com Funcionários</HORA>
 <HORA valor="12.00">Almoço</HORA>
 <HORA valor="15.00">Ligar para cliente</HORA>
 </DIA>
 <DIA numero="D02" tipo="util" semana="quinta">
 <HORA valor="8.00">início do trabalho</HORA>
 <HORA valor="9.00">Visita a Filial</HORA>
 <HORA valor="12.00">Almoço com o Diretor</HORA>
 <HORA valor="13.00">Ligar para cliente</HORA>
 </DIA>
 <DIA numero="D03" tipo="util" semana="sexta">
 <HORA valor="8.00">Início do trabalho</HORA>
 <HORA valor="8.30">Palestra sobre qualidade</HORA>
 <HORA valor="12.00">Almoço com Dimas</HORA>
 <HORA valor="16.00">Vistoriar produtos</HORA>
 </DIA>
 <DIA numero="D04" tipo="nao util" semana="sabado">
 <HORA valor="8.00">Ir ao Mercado</HORA>
 <HORA valor="10.00">Jogar Tênis</HORA>
 <HORA valor="12.00">Almoço com a Esposa</HORA>
 <HORA valor="17.00">Ensaio na Igreja</HORA>
```

Figura 11. Exemplo de atributos no XML. Fonte: FURGERI (2001) pág. 127.

A Figura 12 mostra como o DTD valida o XML

```
<!-- DTD com atributos -->
<!ELEMENT AGENDA (DIA+)>
<!ELEMENT DIA (HORA+)>
<!ELEMENT HORA (#PCDATA)>
<!ATTLIST AGENDA mes CDATA #REQUIRED>
<!ATTLIST DIA
 numero ID #REQUIRED
 tipo CDATA "util"
 semana (segunda|terca|quarta|quinta|sexta|sabado|domingo) #REQUIRED>
<!ATTLIST HORA valor NMTOKEN #REQUIRED>
```

Figura 12. O DTD valida o xml da página anterior. Fonte FURGERI (2001) pág. 128

2.2.3. Web Semântica.

O passo seguinte na evolução do XML e seus DTDs foi a Web Semântica que é, segundo Decker (2000), uma forma de -inserir inteligência e contexto nos códigos XML utilizados para confeccionar páginas Web, com o objetivo de melhorar a interação dos programas com estas páginas e também possibilitar um uso mais intuitivo por parte dos usuários (W3C, 2017).

A Web Semântica faz parte da evolução da “Web de documentos”, onde o próprio W3C (World Wide Web Consortium) ajuda a desenvolver tecnologias para dar suporte à “Web de dados” viabilizando pesquisas como em um banco de dados. O objetivo final é permitir que os computadores executem tarefas mais úteis e que o desenvolvimento de sistemas ofereça suporte às interações na rede. Portanto, “Web Semântica” é a visão do W3C sobre a Web de Dados Conectados. Com a Web Semântica as pessoas têm a possibilidade de criar repositórios de dados na Web, construir vocabulários e escrever regras para interoperar com esses dados (W3C, 2017).

SOUZA e ALVARENGA (2004) acrescentam que “a Web Semântica não é uma Web separada, mas uma extensão da atual”. Na Web Semântica, a informação é disponibilizada com um significado bem definido, propiciando uma “interação entre os computadores e as pessoas”. Desta forma, BERNERS-LEE (2001) e seu grupo de trabalho no W3C definem um planejamento para modificar a Web

2.3. ENSINO DE LINGUAGENS ATRAVÉS DE JOGOS

Segundo COSTA (2016), *"Games* podem atuar como coadjuvantes no desenvolvimento de funções cognitivas e sensoriais, melhorando a noção espacial, as habilidades motoras, o processo de tomada de decisões e a autonomia".

2.3.1. Serious Games

De acordo com MODESTO e SCAVACINI (2013), o conceito de Serious Games é utilizado para descrever games com um propósito específico, que vão além do entretenimento e oferecem outras experiências, como educação e treinamento. Essa classe de jogos usa a simulação de situações práticas para oferecer treinamento para profissionais, situações críticas em empresas, conscientização para crianças, jovens e adultos e até para situações bem corriqueiras, como escolher a cor de um carro.

GEE (2009) afirma que o jogo cria condições para a participação da criança, "principalmente nos momentos em que as crianças fazem uso dos jogos, criam sua identidade, interagem, produzem, arriscam-se e formulam hipóteses para suas resoluções".

2.3.2 Exemplos de jogos para aprendizagem de linguagens

Dentre os jogos utilizados na aprendizagem de linguagens, podemos citar como exemplo o Touch Develop (Figura 13), disponibilizado pela Microsoft Research de forma gratuita para que o jogador aprenda a criar aplicativos para celular, *tablet* ou *laptop*. Esse jogo funciona em diferentes sistemas operacionais (IOS, Android, Windows Mac e Linux), oferece orientações simples e qualquer pessoa, desde uma criança pequena até um adulto sem conhecimento prévio, pode aprender de forma lúdica, sem as pressões do ensino tradicional.

A Figura 13 exibe a tela de um dos jogos pesquisados.

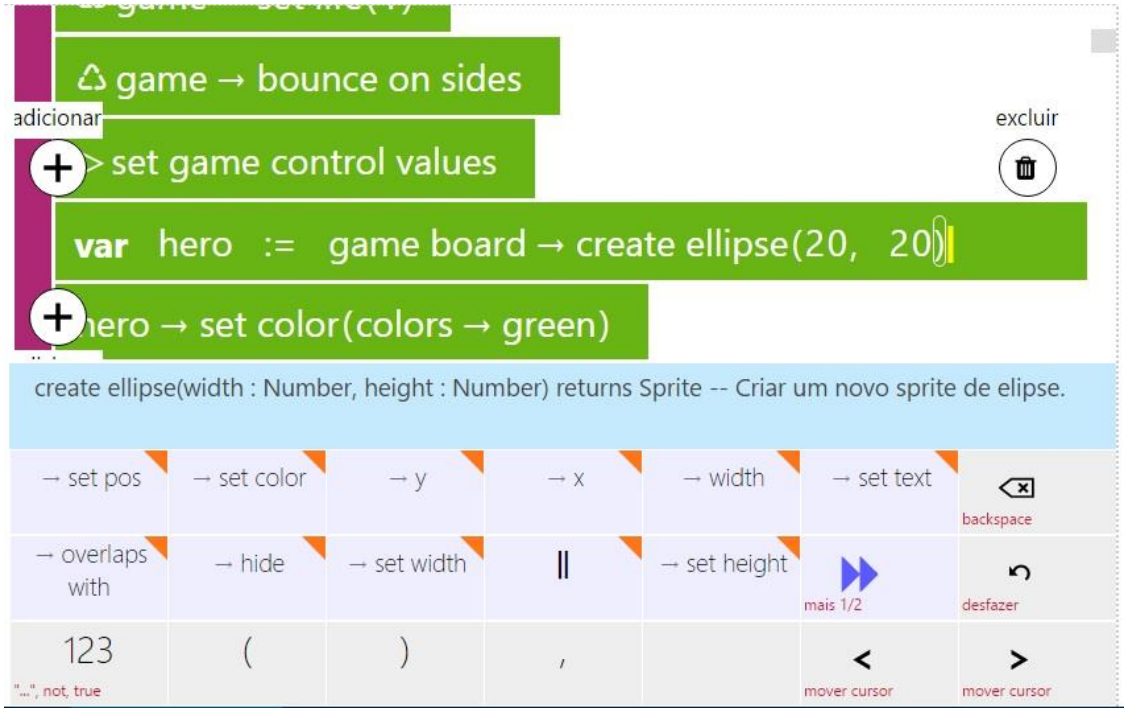


Figura 13. Screenshot do jogo Touch Develop. Fonte:https://www.touchdevelop.com/.

Em uma das modalidades, a imagem de uma tartaruga vai desenhando uma linha por onde ela passa (Fig. 14).

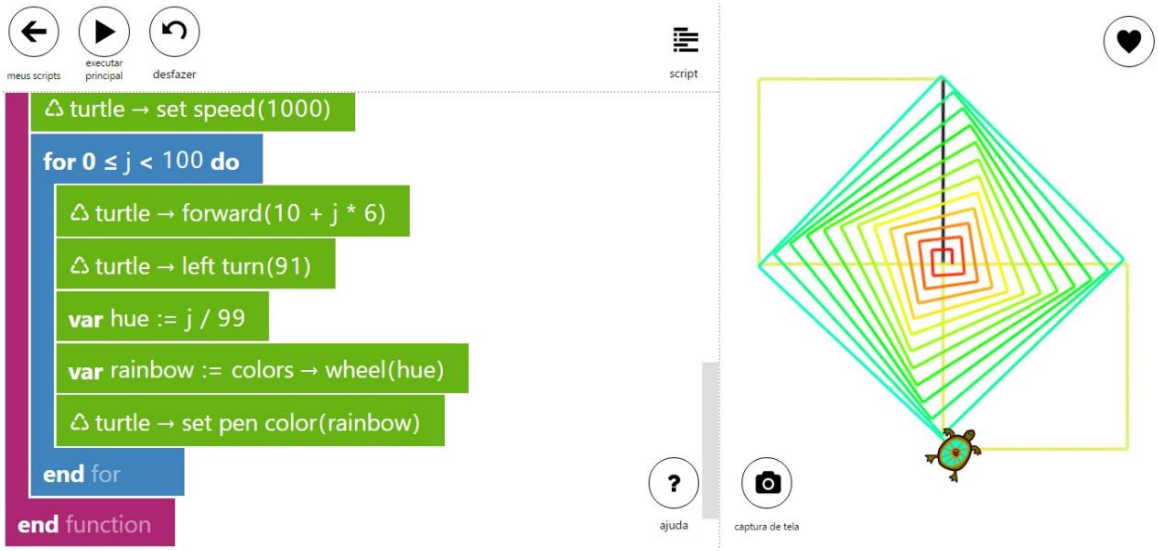


Figura 14. Screenshot do jogo Touch Develop - Turtle. Fonte:https://www.touchdevelop.com/. Acesso em 03/maio/2017

Outro exemplo de jogo para o aprendizado de programação é o Alice (Fig. 15)

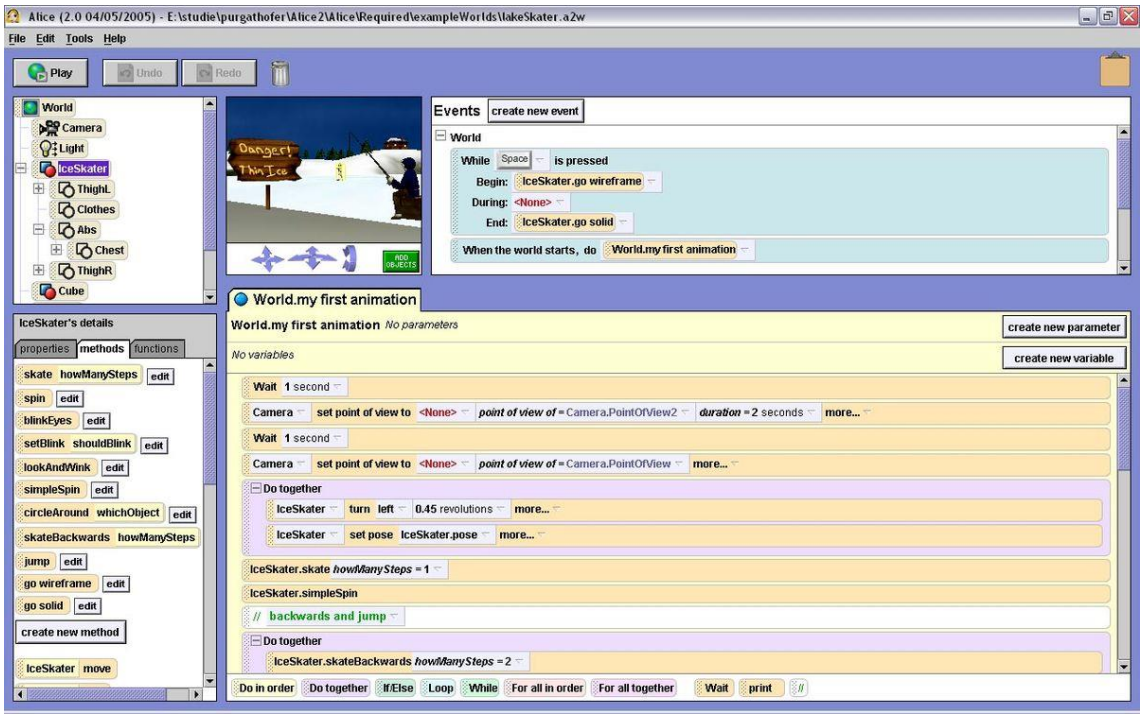


Figura 15. Software Alice, para aprendizado de programação orientada a objetos e criação de jogos. Fonte: <https://upload.wikimedia.org/wikipedia/commons/thumb/5/57/Alice-2-screenshot.jpg/1200px-Alice-2-screenshot.jpg>. Acesso em: 28/09/2017

Para o ensino da linguagem de marcação HTML, existe por exemplo, o curso básico apresentado pelo W3C, que não chega a ser um jogo, mas é bem interativo e didático (Figuras 16 e 17).

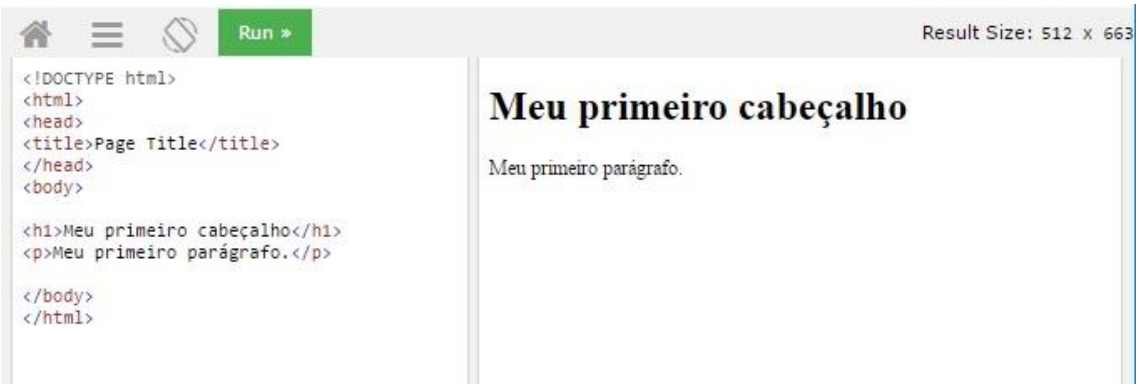


Figura 16. Curso de HTML Básico. W3Schools. Fonte: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_intro

Esse curso é gratuito, totalmente traduzido para o português e nele é possível digitar o código do lado esquerdo e ver o resultado das modificações feitas ao código, do lado direito, começando do zero até chegar ao nível mais complexo e criar um site completo.

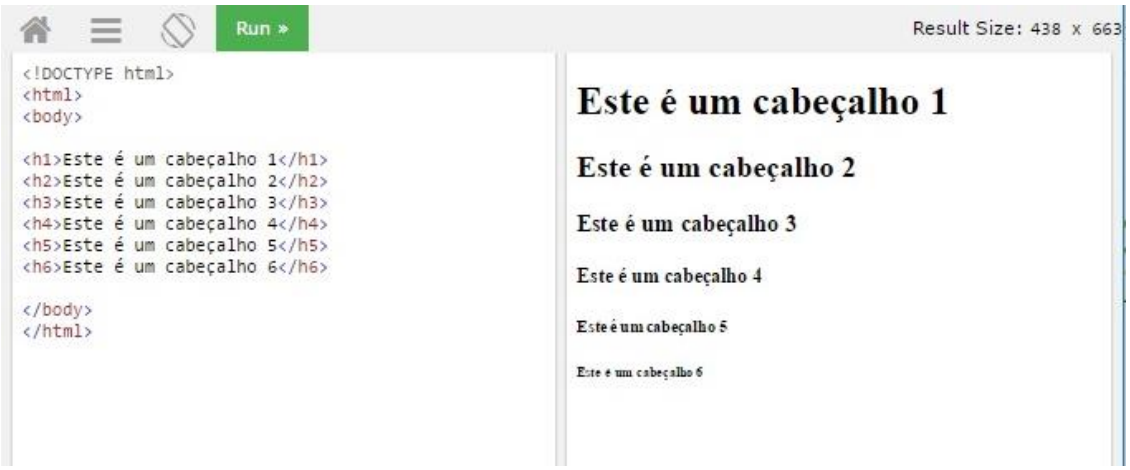


Figura 17. Curso de HTML Básico. W3Schools. Fonte: https://www.w3schools.com/html/tryit.asp?filename=tryhtml_basic_headings

Um artigo publicado no site Medium.com, com o título "*Text is keeping kids from coding*" ("O texto está afastando as crianças da programação", em tradução livre), escrito por DANNY YAROSLAVSKI (2017), explica que foi realizada uma pesquisa sobre um jogo para aprendizado de programação, onde o texto do código era bem simplificado por se tratar de um jogo para crianças de 5 anos, sendo totalmente projetado em linguagem coloquial, mas mesmo assim, as crianças e os pais delas acharam o texto complicado e que o jogo era mais adequado para a faixa etária de 9 a 10 anos.

A conclusão dessa pesquisa foi que o texto era intimidador, portanto trocaram as palavras por ícones e repetiram a experiência.

Desta vez as crianças, e os pais delas, acharam o jogo bem mais fácil e divertido e consideraram que ele era adequado para a faixa etária a partir de 5 anos.

Considerando os resultados dessa pesquisa, o modelo proposto neste trabalho utilizará mais imagens do que texto.

2.4. METODOLOGIA PARA ANÁLISE DO MODELO PROPOSTO.

2.4.1 Padrão de arquitetura MVC.

A arquitetura MVC é uma proposta para a modelagem e implementação de sistemas orientados a objetos que permite organizar todos os componentes em três camadas:

Modelo, **V**isão e **C**ontrole (MVC) (SOUZA, 2017):

- Modelo é a camada de dados. Na maioria dos sistemas, é gerenciada por um sistema relacional de dados através de um sistema gerenciador de banco de dados. Nesta camada está a maioria das regras de negócio da aplicação, convencionalmente implementada por frameworks como o JPA ou Hibernate.
- Visão é a camada que gerencia a exibição dos dados na tela.
- Controle – ou Controlador – é a camada que liga as ações da interface com a camada de modelo.

A arquitetura de *software* de um sistema computacional pode ser definida como a sua estrutura, que é composta de elementos de *software*, as propriedades externamente visíveis desses componentes, e o relacionamento entre eles. Ou seja, a arquitetura define os elementos de *software* e como eles interagem entre si. O padrão de arquitetura MVC (*Model-View-Controller*) é um dos mais antigos e mais utilizados atualmente. (DevMedia, 2017)

É utilizado o padrão MVC porque é um verdadeiro padrão de projeto de software, testado e comprovado, que transforma uma aplicação em pacotes de desenvolvimento rápido, de fácil manutenção e modular.

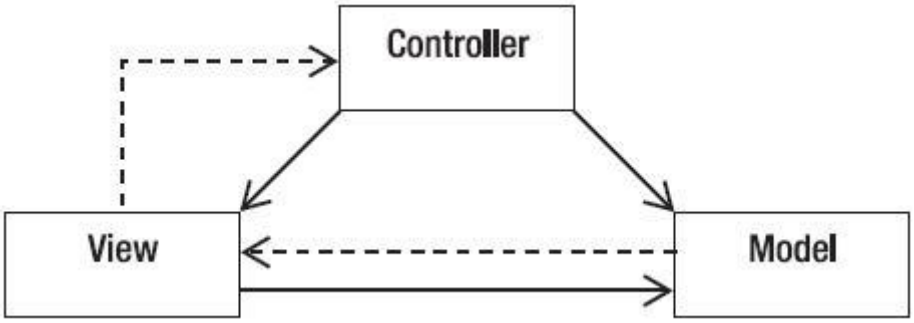


Figura 18. Objetos utilizados no MVC e suas interações. Fonte: <http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308>

No padrão MVC, primeiramente o controlador (*Controller*) “ouve” os eventos das interfaces de entrada do mouse, teclado ou tela de touchscreen (View) e mapeia essas ações dos usuários em comandos que são enviados para a camada modelo (Model), responsável pela manutenção das informações persistentes e de suas regras de negócio. A camada de modelo (*Model*) gerencia um ou mais elementos de dados, responde a perguntas sobre o seu estado e responde a instruções para mudar de estado. A camada de modelo sabe o que o aplicativo quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema que estamos tentando resolver. Por fim, a visão (*View*) gerencia a área retangular do display e é responsável por apresentar as informações para o usuário através de uma combinação de gráficos e textos. A visão não sabe nada sobre o que a aplicação está atualmente fazendo, tudo que ela realmente faz é receber instruções do controle e informações do modelo e então exibi-las. A visão também se comunica de volta com o modelo e com o controlador para reportar o seu estado. (DEVMEDIA, 2017)

2.4.2. A linguagem UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) e seus diagramas.

Linguagem criada para evitar conflitos de definições e nomenclaturas na área de modelagem, a UML é uma linguagem padrão para visualização, especificação, construção e documentação de um aplicativo ou projeto de software. Esta linguagem permite “desenhar” uma “planta” do sistema e ter uma visão lógica, com maior clareza de todas as etapas do projeto para facilitar a implementação do mesmo (Revista SQL Magazine, set-2017). No entanto, essa linguagem não especifica uma metodologia para a modelagem de sistemas.

Apesar de a UML definir muitos diagramas, para este TCC usaremos somente os diagramas UML para: caso de uso, diagrama de classes e diagrama de sequências. Para a modelagem do sistema proposto neste TCC, usaremos a metodologia aplicada na Unidade Curricular de Processos de Desenvolvimento de Sistemas, onde são descritos para todos os casos de uso: a) Uma interface de operação com o nome e os atores; b) O fluxo de operação dessa interface; c)

Um diagrama de caso de uso; e) Um diagrama de classe do caso de uso no modelo MVC e, d) Um diagrama de sequências. (SOUZA, 2017)

2.4.2.1 - Diagrama de casos de uso.

Um caso de uso representa uma interação de um usuário/ator do sistema (pessoa, outro sistema, robô, etc.) com uma funcionalidade do sistema (caso de uso).

> Os casos de usos podem ser aplicados para captar o comportamento pretendido do sistema que está sendo desenvolvido, sem ser necessário especificar como esse comportamento é implementado. Os casos de uso fornecem uma maneira para os desenvolvedores chegarem a uma compreensão comum com os usuários finais do sistema e com os especialistas do domínio. Além disso, os casos de uso servem para ajudar a validar a arquitetura e para verificar o sistema à medida que ele evolui durante seu desenvolvimento. À proporção que você implementa o seu sistema, esses casos de uso são realizados por colaborações cujos elementos trabalham em conjunto para a execução de cada caso de uso. (BOOCH, 2005).

O diagrama comportamental de caso de uso em UML é representado por uma elipse e o ator é representado por uma figura de uma pessoa. Os casos de usos podem estar relacionados através de associações do tipo: associação simples, agregação, composição, extensão, inclusão e herança. Agregação é um tipo de associação que representa uma relação, é parte de um todo. Embora tanto a parte como o todo possam existir isoladamente, por exemplo: um time e os jogadores, cada jogador existe independente do time existir. Já na composição, o objeto parte só existe dentro do todo. Um exemplo de composição são os andares de um edifício. Os andares não podem existir independente do edifício. A extensão significa que um caso de uso estende outro, ou seja opcionalmente ele pode vir a ser executado. No caso da inclusão, o caso de uso incluído obrigatoriamente será executado após o caso de uso principal. Dizemos que um caso de uso herda outro quando este recebe todas as funcionalidades prontas,

implementado somente aquelas inexistentes no caso de uso herdado. (SOUZA, 2017)

2.4.2.2 - Diagrama de classes.

Uma classe é uma abstração de um conjunto de objetos. Como tal, as classes são compostas de três elementos: nome da classe, atributos e métodos. Os atributos definem os dados dos objetos e os métodos suas ações.

O diagrama estrutural de classes representa as entidades (objetos, telas, controladores, etc.) e suas associações. Tem as mesmas associações dos casos de uso, exceto a inclusão e extensão. Na modelagem do sistema proposto, usaremos o diagrama de classes para representar o modelo MVC aplicado ao caso de uso e, nesse caso, sempre teremos as classes de interface, de controle e de modelo (dados). Também usaremos o diagrama de classes para representar as associações entre as classes de entidades de banco de dados. (SOUZA, 2017)

2.4.2.3 - Diagrama de sequências.

O Diagrama comportamental de sequências reflete a execução do caso de uso (fluxos) e baseia-se na interação entre os objetos das classes do diagrama de classes do caso de uso. Sua representação é feita através de trocas de mensagens entre os objetos numa linha de tempo que reflete o fluxo definido no caso de uso correspondente. (SOUZA, 2017)

Modelo

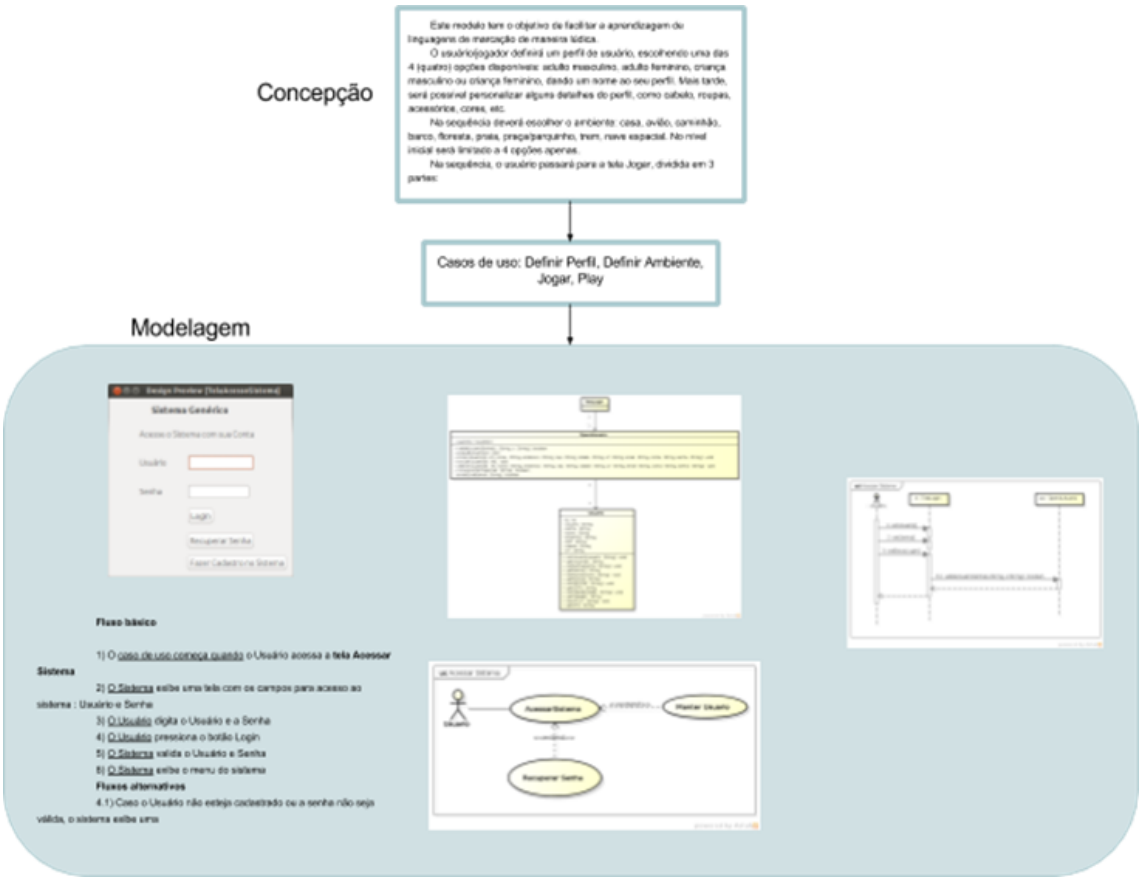


Figura 19. Estrutura do Modelo proposto

3. DESENVOLVIMENTO DO TRABALHO – ESPECIFICAÇÃO DO MODELO

A UML não descreve uma metodologia para modelagem de um sistema. Para tanto, aplicamos a metodologia sugerida na Unidade de Processos de Desenvolvimento de Sistemas, como descrita abaixo (SOUZA, 2017):

- Descrição textual do problema
- Listar os requisitos funcionais (objetivos)
- Descrever os Casos de Uso (CDU)
- Desenhar uma tela para cada Caso de Uso
- Desenhar o Diagrama de Caso de Uso
- Listar as entidades e seus atributos
- Desenhar o Diagrama de Classes
- Desenhar os Diagramas de Sequência

3.1. DESCRIÇÃO TEXTUAL DO MODELO BASEADO NO PROBLEMA (ITEM 1.2).

Nome do modelo: *XML-Kids*. Consiste em um jogo para facilitar a aprendizagem da linguagem de marcação XML de maneira lúdica.

O usuário/jogador/ator define um perfil, escolhendo uma das 4 (quatro) opções disponíveis: adulto masculino, adulto feminino, criança masculino ou criança feminino, dando um nome ao seu perfil. Mais tarde, será possível personalizar alguns detalhes do perfil, como cabelo, roupas, acessórios, cores, etc.

Na sequência deverá escolher o ambiente: casa, avião, caminhão, barco, floresta, praia, praça/parquinho, trem, nave espacial. No nível inicial será limitado a 4 (quatro) opções apenas.

Na sequência, o usuário passará para a tela Jogar, dividida em 3 (três) partes:

A parte superior terá um ícone indicando o ambiente escolhido, o botão Play e uma barra horizontal com vários menus de objetos. Por exemplo, se o ambiente escolhido for a casa, os menus de objetos terão opções de móveis, eletrodomésticos, tapetes, acessórios, etc; se for escolhido o ambiente de avião, os menus de objetos serão referentes a tudo que possa ter em um avião, e assim por diante. Um exemplo de ambiente pode ser visto na Figura 20.

Figura 20. Exemplo de Ambiente Casa. Fonte: https://pt.depositphotos.com/

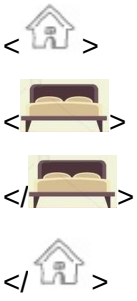
O sistema grava um arquivo de ações (log) que pode ser acionado através de um botão de play. O log deverá gravar todas as ações do ator (objeto movimentado, alteração do XML, alteração de atributos de objetos, etc.), bem como o arquivo XML gerado. O log sempre será gerado e poderá servir para analisar como os usuários resolvem as situações propostas nesse ambiente. A opção play exibirá uma animação com todos os passos executados pelo usuário.

A parte central será o setor de jogo, que representará o ambiente escolhido, dentro do qual o usuário colocará os objetos arrastados da parte superior.

Na parte inferior será exibido o código XML que refletirá tudo que o usuário fizer no setor de jogo, e ao mesmo tempo, o usuário poderá editar o código para modificar atributos dos objetos colocados no setor de jogo.

O código XML terá apenas desenhos (ícones) para crianças não alfabetizadas.

Exemplo:



Para crianças já alfabetizadas, terá desenhos e, também, do lado do desenho, terá o texto do próprio código.

Exemplo:

< casa>

< sofá>

</ sofá>

</ casa>

O mesmo acontecerá com os atributos usando bandeirinhas coloridas do lado do nome da cor. Uma seta grande e brilhante indicará que pode clicar em uma setinha pequena do lado da bandeirinha e ao clicar poderá escolher mais bandeirinhas de outras cores. Para os tamanhos serão utilizados ícones grandes e pequenos. E haverá distinção visual entre tag, atributo e conteúdo.

Quando o usuário arrastar um objeto da barra superior para a parte central, a parte inferior exibirá o código correspondente àquela ação.

Após arrastar todos os objetos para a parte central, a gravação das ações será interrompida, a cena ganhará animação e aparecerão balões acompanhados de música para comemorar.

Ao clicar no botão Play, uma animação mostrará toda a sequência das ações realizadas pelo usuário até completar a cena.

O Jogo será composto por um conjunto de peças que caracterizam um ambiente. O ambiente sempre tem uma peça inicial que conterá as demais. As peças possuem uma hierarquia, por exemplo: sempre que o usuário escolhe uma peça, automaticamente é criado um arquivo.

3.2. REQUISITOS FUNCIONAIS

Os requisitos funcionais são todas as funcionalidades que o sistema deverá ter sem especificar como será implementado. Os requisitos são obtidos a partir da descrição do sistema e são ações identificadas por verbos no infinitivo.

- Identificar perfil do usuário
- Configurar perfil do usuário
- Escolher ambiente para o jogo

- Jogar
- Personalizar objetos
- Configurar/Renderizar os objetos
- Analisar os logs de ações dos usuários. (poderemos identificar qual o nível de dificuldade encontrado na resolução dos ambientes).

3.3. CASOS DE USO

Caso de uso é a maneira de descrever como será o uso de uma funcionalidade de um sistema para atender um ator. Essa descrição segue um padrão específico, de maneira que o leitor do caso de uso – o programador do sistema ou o usuário que utilizará o caso de uso – entenda a funcionalidade de uma maneira única e que facilite sua compreensão por qualquer interessado pelo sistema. Ele serve para validar e facilitar modificações tanto pelos analistas quanto pelos usuários do sistema. Os casos, na maioria das situações, agregam um conjunto de requisitos.

Identificamos os seguintes casos de uso conforme a descrição do problema e do modelo proposto.

1. Definir Perfil e Ambiente
2. Jogar
3. Personalizar objetos
4. Analisar logs dos usuários

Na sequência serão descritos os principais casos de uso conforme a metodologia adotada, e que são prioritários para o desenvolvimento futuro de um sistema.

3.3.1. Caso de Uso Definir Perfil e Ambiente

Nome do Sistema:	XMLKids
Nome do Caso de Uso:	***Definir Perfil***
Ator(es):	Usuário

Na Figura 21 é exibido um exemplo de como será a tela do caso de uso Definir Perfil e Ambiente

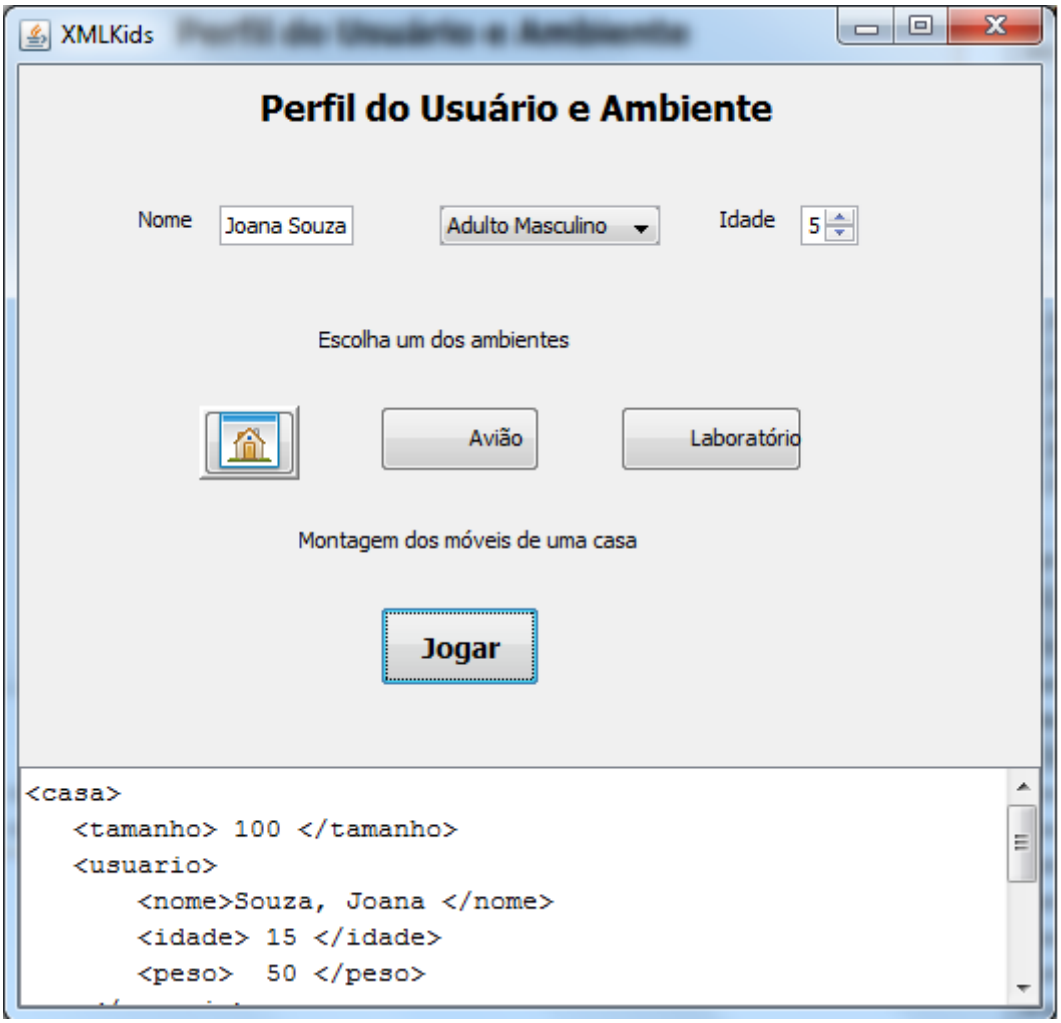


Figura 21. Tela Perfil do Usuário e Ambiente, com o código XML na parte inferior

Fluxo básico

1) O caso de uso começa quando o Usuário acessa a tela **Perfil do Usuário e Ambiente**

2) O Usuário informa o nome

3) O Usuário informa a idade

4) O Sistema localiza os ambientes disponíveis para a idade indicada

5) O Sistema exibe uma tela com os perfis e os XMLs correspondentes

6) O Usuário escolhe um Ambiente

7) O Sistema exibe o código XML para aquele perfil definido

8) O Usuário clica no botão jogar

9) O Sistema valida os dados

10) O Sistema grava as informações

Fluxos alternativos

3.1) A idade deve ser maior ou igual a 1 (um) ano.

7.1) O Código em XML é editável. Caso o usuário queira, poderá alterar diretamente o arquivo em XML. Neste caso, o sistema deverá validar o XML apontando onde há erros de sintaxe e de semântica da estrutura do XML.

9.1) Caso o usuário não informe um nome, o sistema solicitará a informação novamente.

10.1) Quando o usuário clicar no botão Jogar, o sistema acionará o caso de uso Jogar.

3.3.1.1 Diagrama do Caso de Uso Definir Perfil e Ambiente

A Figura 22 mostra o diagrama deste caso de uso.

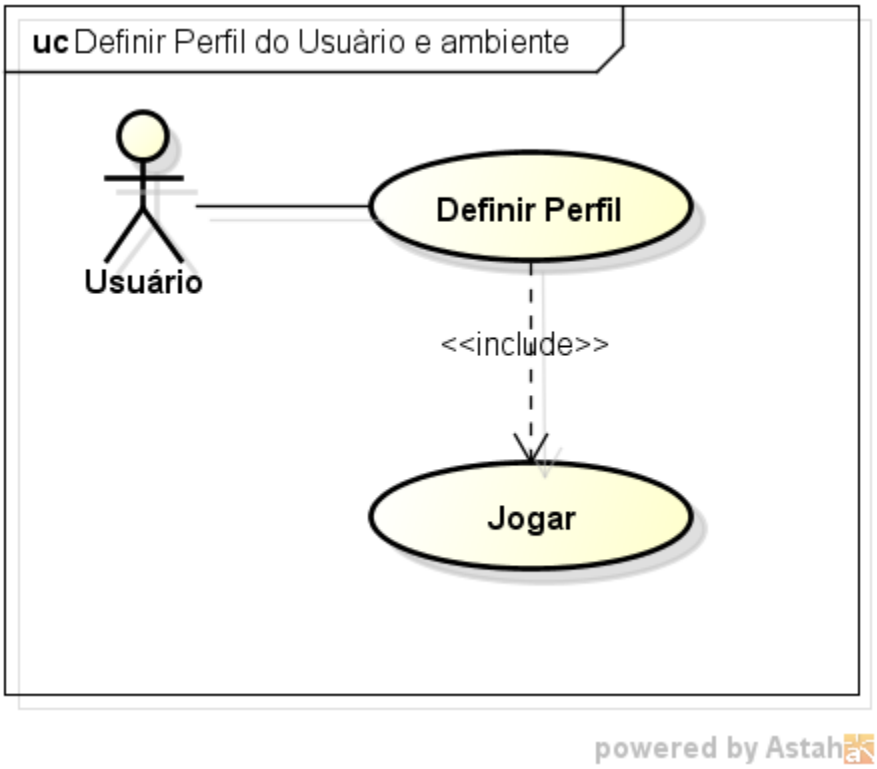


Figura 22. Diagrama do Caso de uso Definir Perfil

3.3.1.2 Diagrama de classes do CDU Definir Perfil e Ambiente em MVC

A Figura 23 mostra o diagrama de classes deste caso de uso.

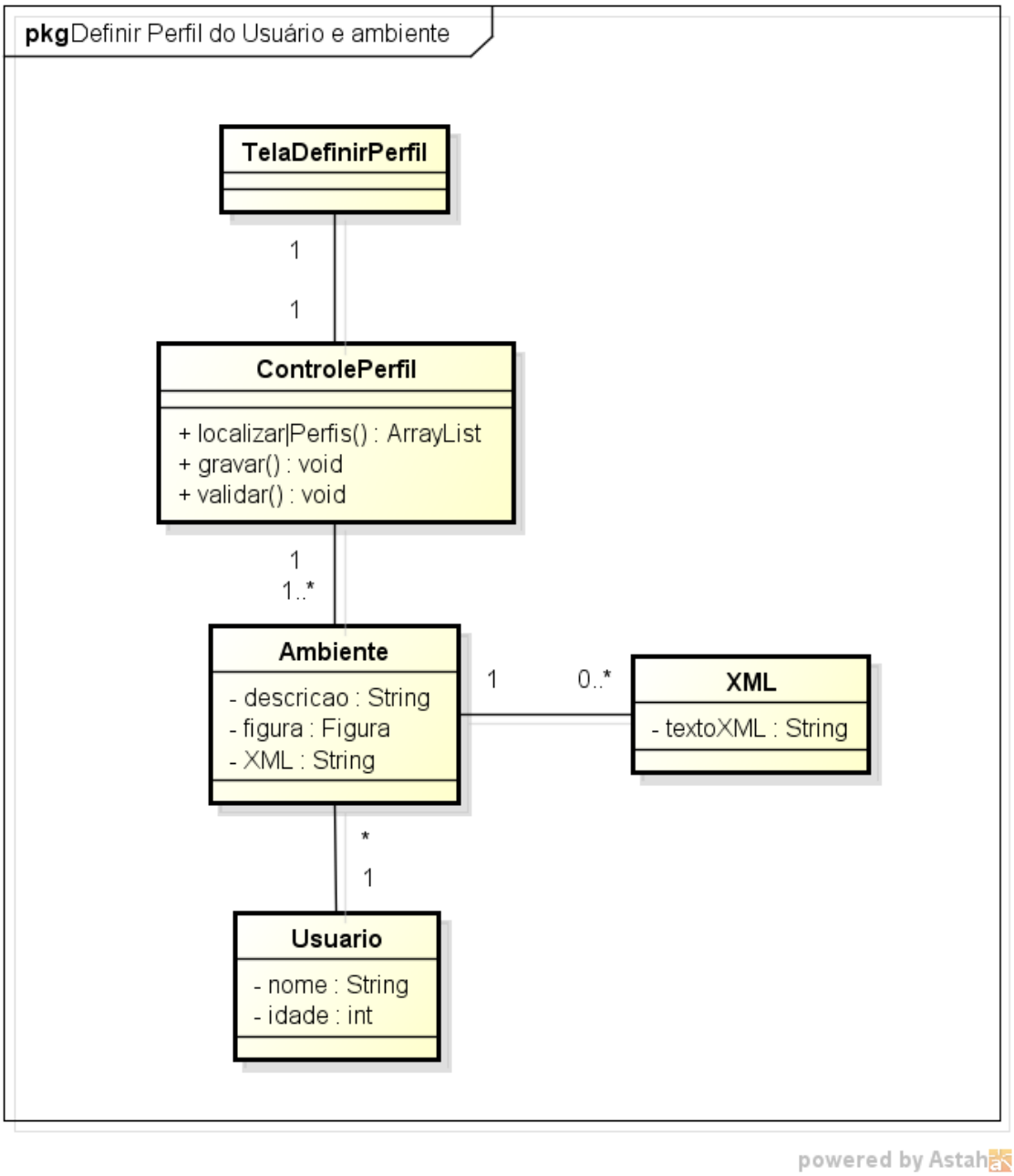


Figura 23. Diagrama de Classes do Caso de Uso Definir Perfil do Usuário e Ambiente em MVC

3.3.1.3 Diagrama de Sequência do Caso de Uso Definir Perfil e Ambiente

A Figura 24 mostra o diagrama de sequência deste caso de uso.

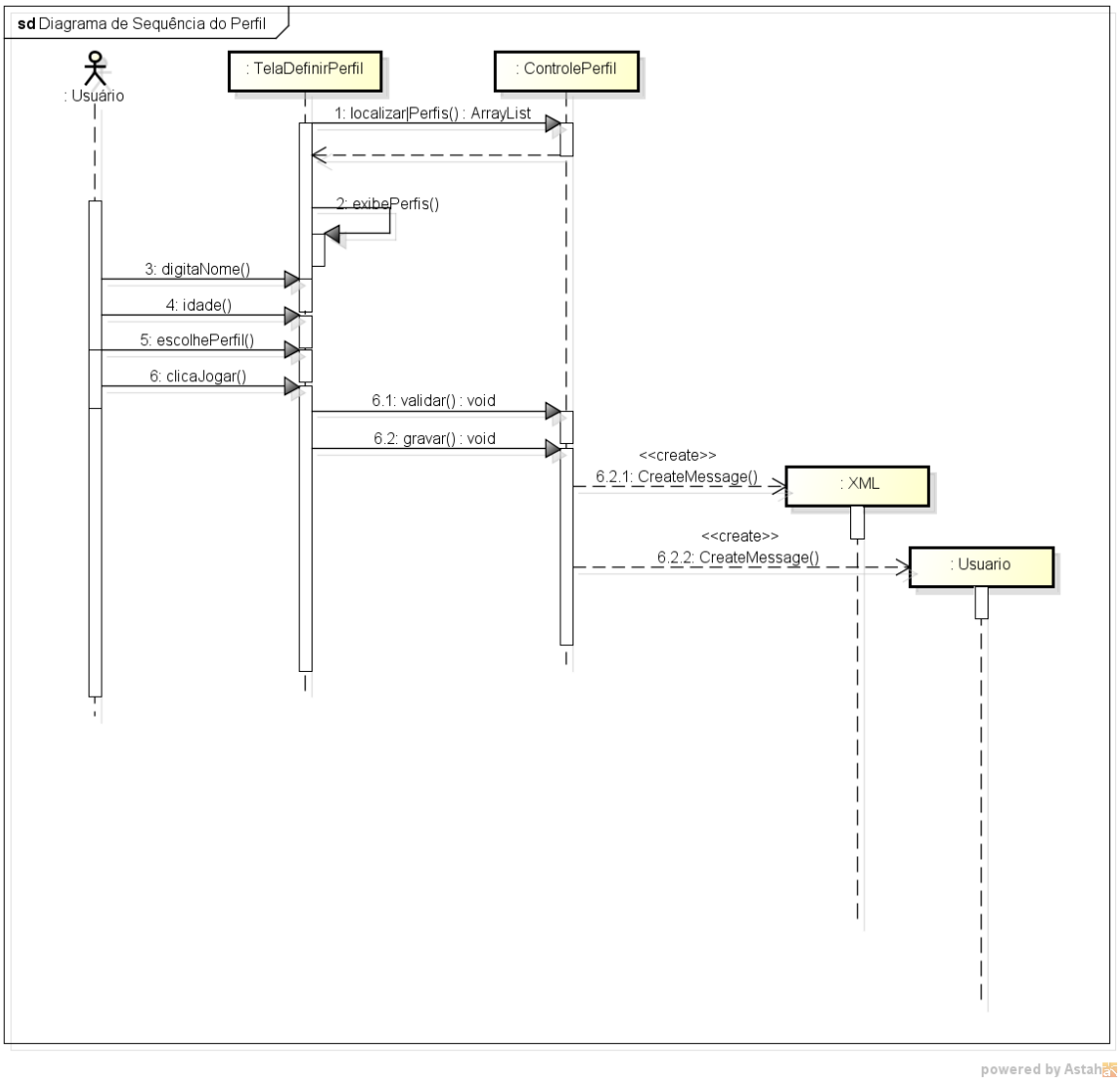


Figura 24. Diagrama de sequência do Caso de Uso Definir perfil do usuário e ambiente

3.3.2. Caso de Uso Jogar

Nome do Sistema:	XMLKids
Nome do Caso de Uso:	***Jogar***
Ator(es):	Usuário
Pré-condição:	Ambiente e perfil definidos

A Figura 25 mostra a tela deste Caso de Uso.

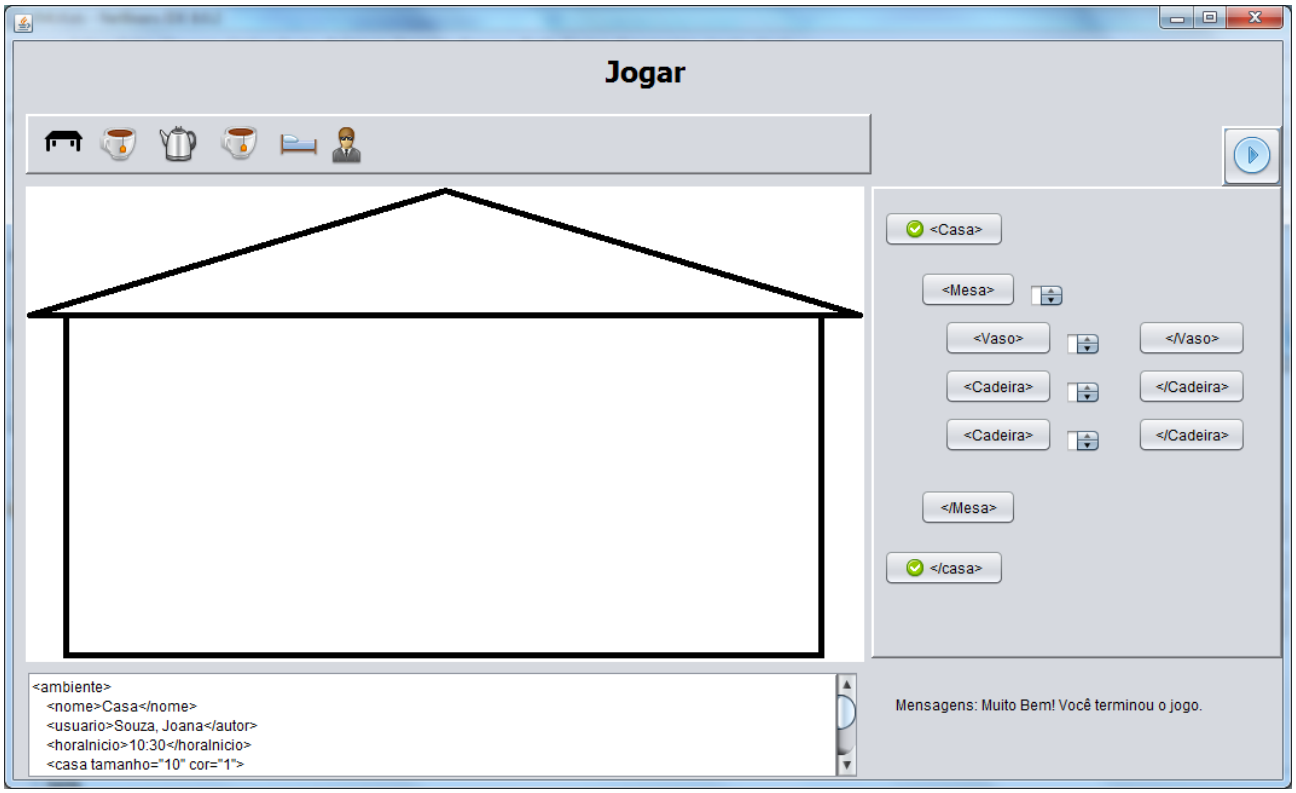


Figura 25. Tela do caso de uso Jogar. Na parte central da tela serão inseridos os objetos que se encontram na parte superior. Na parte inferior da tela é exibido o código XML

Fluxo Principal

1) O Caso de Uso começa quando o Usuário acessa a tela Jogar

2) O Sistema localiza o ambiente e obtêm objetos do jogo

3) O sistema exibe na parte central da tela o ambiente escolhido (exemplo: casa, avião, praia, etc.) com uma paleta de objetos e um texto em XML.

4) O Usuário arrasta um objeto do menu de objetos (topo da tela), para a figura que caracteriza o ambiente, encaixando o objeto no lugar específico.

5) O Sistema verifica se o objeto combina com outro objeto

6) O Sistema exibe o código XML na parte inferior da tela e na lateral direita.

7) O sistema grava a ação do usuário.

Fluxos alternativos

3.1) Se o usuário clicar no botão de fechar, o sistema volta para o caso de uso Definir Perfil.

5.1) Caso o objeto seja colocado em um lugar não adequado, este é automaticamente devolvido para o menu de objetos e um aviso é lançado.

5.2) Quando o objeto é colocado na posição correta, o sistema emite uma saudação e exibe na parte inferior da tela o correspondente em XML.
5.3) Quando todos os objetos tiverem sido colocados corretamente, o sistema emite uma saudação de finalização.
5.4) Caso o Usuário clique no botão Play, será acionado o caso de uso Play - reproduz as ações executadas pelo usuário.
5.5) Sempre que o usuário movimenta um objeto, o sistema grava um log com ação executada.
5.6) Caso o usuário opte por editar o XML (tela inferior), somente os atributos dos objetos já estacionados poderão ser alterados (ex.: cor, tamanho,
<casa [verde, vermelho, azul] [pequeno médio grande]>
Isto também pode ser feito na lateral direita da tela.

3.3.2.1 Diagrama do Caso de Uso Jogar

A Figura 26 mostra o diagrama deste caso de uso.

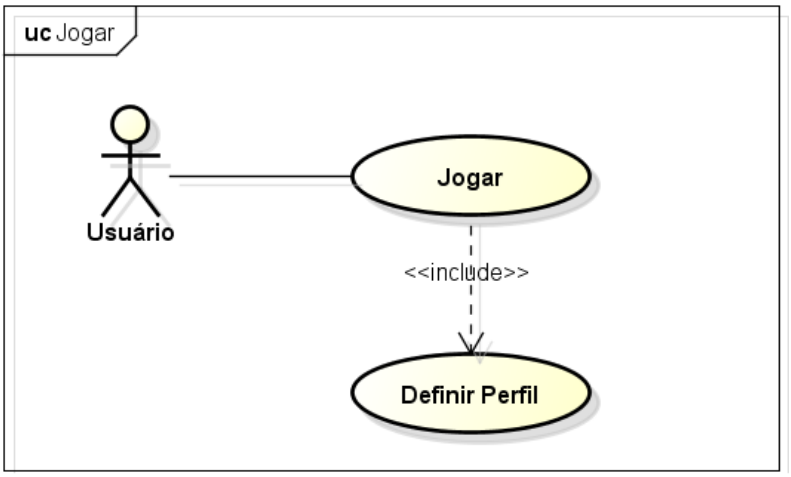


Figura 26. Diagrama do Caso de Uso Jogar

3.3.2.2 Diagrama de classes do CDU Jogar em MVC

A Figura 27 mostra o diagrama de classes deste caso de uso.

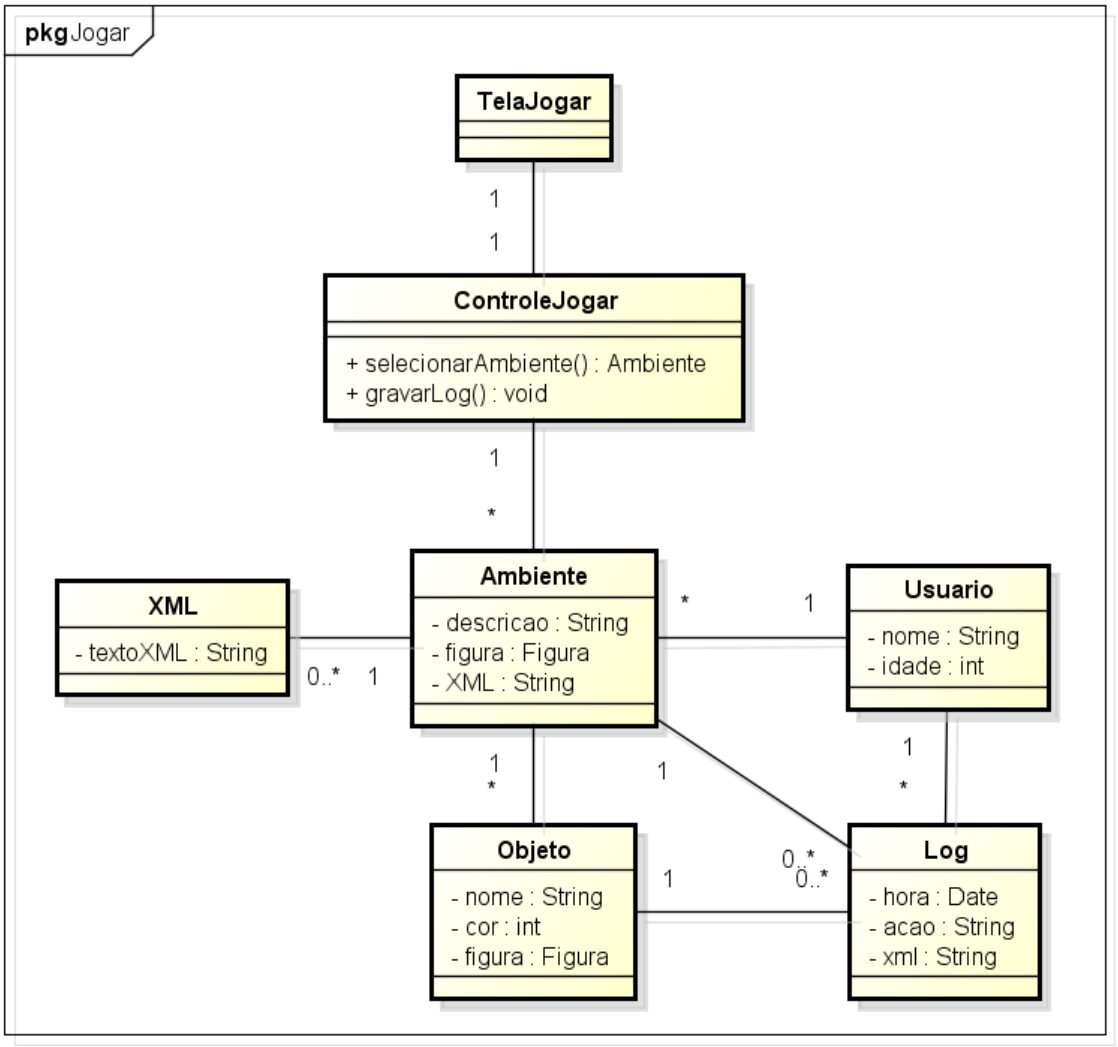


Figura 27, Diagrama de classes do CDU Jogar em MVC

3.3.2.3 Diagrama de Sequência do CDU Jogar

A Figura 28 mostra o diagrama de sequência deste caso de uso.

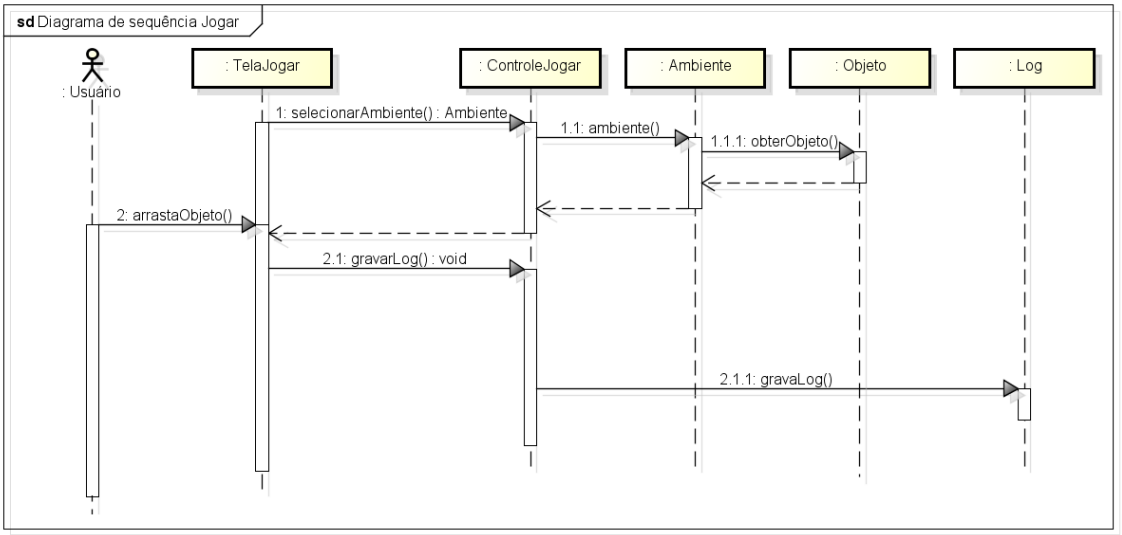


Figura 28. Diagrama de Sequência do CDU Jogar

3.3.3. Caso de Uso Personalizar objetos

Nome do Sistema:	XMLKids
Nome do Caso de Uso:	***Personalizar objetos***
Ator(es):	Usuário

A Figura 29 mostra a tela deste caso de uso

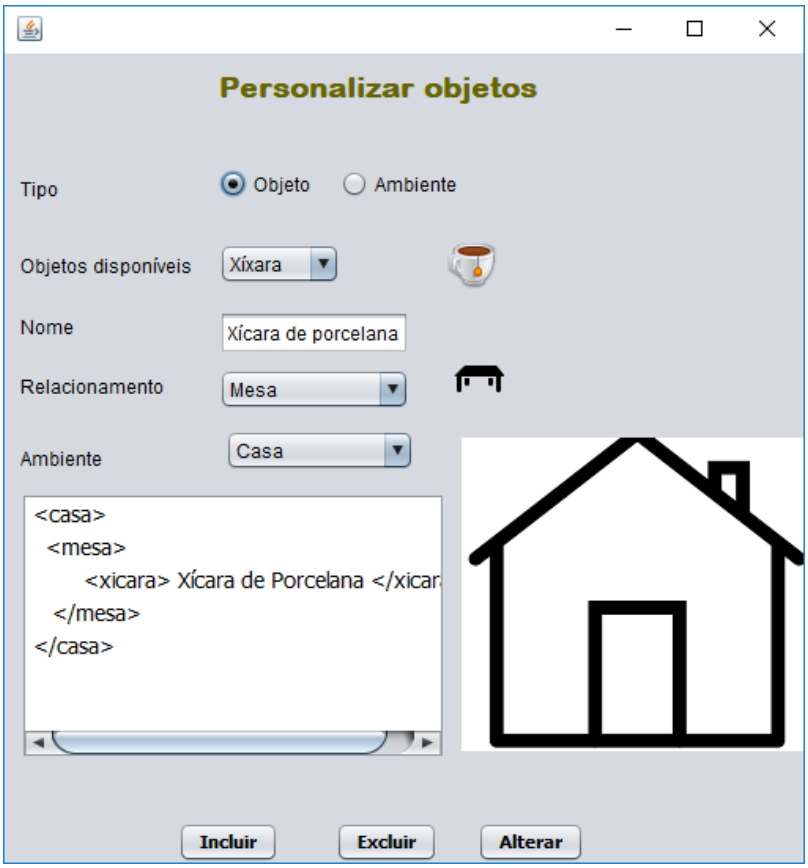


Figura 29. Tela Caso de Uso Personalizar objetos

Fluxo Principal

1) O Caso de Uso começa quando o Usuário acessa a tela Personalizar objetos

2) O ator escolhe o tipo de objeto: Ambiente ou objeto

3) O sistema exibe uma lista com os objetos/ambientes disponíveis

4) O ator seleciona um objeto/ambiente

5) O Sistema exibe o nome e o relacionamento

6) O ator digita o nome do objeto

7) O ator informa a relação do objeto com outro objeto

8) O ator informa o ambiente

9) O ator clica no botão Incluir/Excluir/Alterar

10) O Sistema faz a consistência dos dados do objeto

11) O Sistema altera o XML do ambiente para comportar o objeto

12) O sistema exibe o XML com as alterações

13) O sistema inclui o objeto

Fluxo Alternativo

7.1) Caso o sistema detecte alguma inconsistência com os dados do objeto, será emitido uma observação e um pedido para fazer a correção.

3.3.3.1 - Diagrama do caso de uso Personalizar Objetos

A Figura 30 mostra o diagrama deste caso de uso.

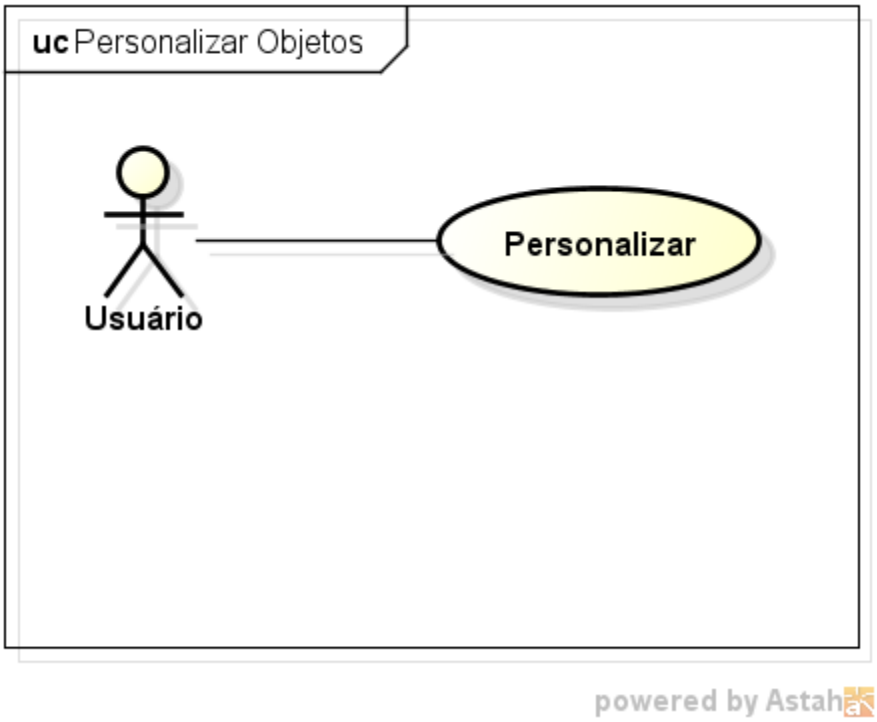


Figura 30. Caso de Uso Personalizar objetos

3.3.3.2 Diagrama de Classes do caso de uso Personalizar Objetos

A Figura 31 mostra o diagrama de classes deste caso de uso.

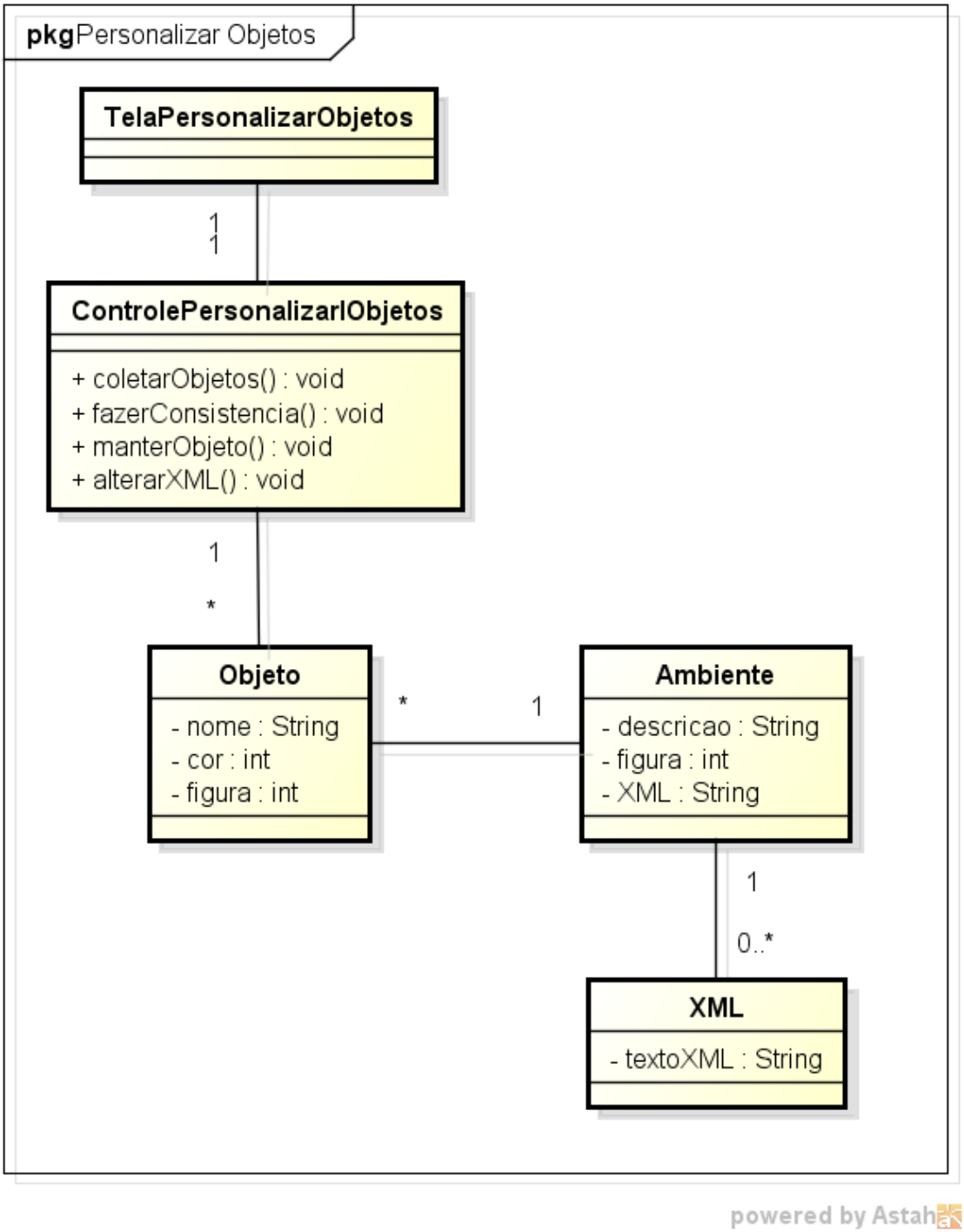


Figura 31. Diagrama de Classes do caso de uso Personalizar objetos

3.3.3.3 Diagrama de sequências do caso de uso (Figura 32)

A Figura 32 mostra o diagrama de sequência deste caso de uso.

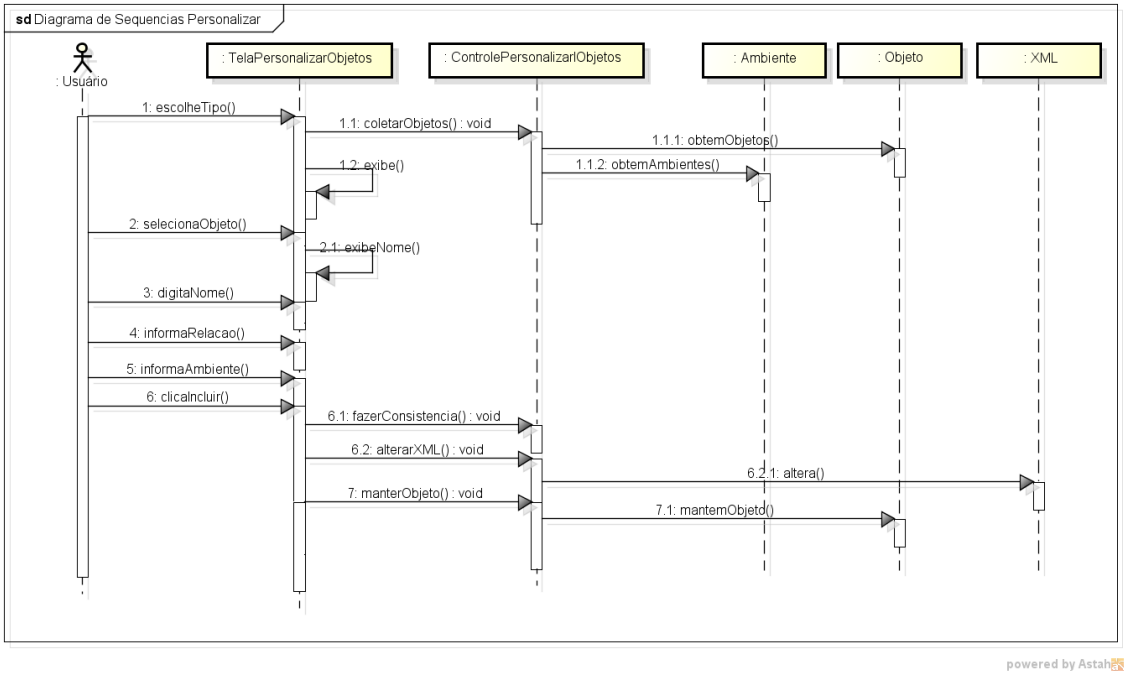


Figura 32. Diagrama de sequências do caso de uso Personalizar objetos

3.3.4. Diagrama de classes de entidades de banco de dados.

As classes de entidades são muito semelhantes às tabelas de um banco de dados, exceto por terem seus próprios métodos e poderem implementar as funcionalidades da orientação a objetos. No diagrama abaixo apresentamos apenas as classes com suas propriedades e sem os seus métodos, além de suas associações. A razão porque não incluímos os métodos deve-se ao fato desses serem padronizados e próprios dos frameworks adotados. Como a maioria dos sistemas de armazenagem de dados é baseada no modelo relacional, geralmente, quando a implementação é feita, há a necessidade de conversão do modelo relacional para o modelo orientado a objetos. Esse processo é denominado Mapeamento Objeto Relacional. Um diagrama de classes de entidades também pode ser utilizado para gerar um banco de dados no modelo relacional (SOUZA, 2017)

Obtemos as entidades a partir da descrição textual do problema que queremos modelar. A técnica consiste em identificar os principais substantivos e suas propriedades.

Abaixo temos um diagrama representando todas as entidades e suas relações para o sistema proposto (Figura 33).

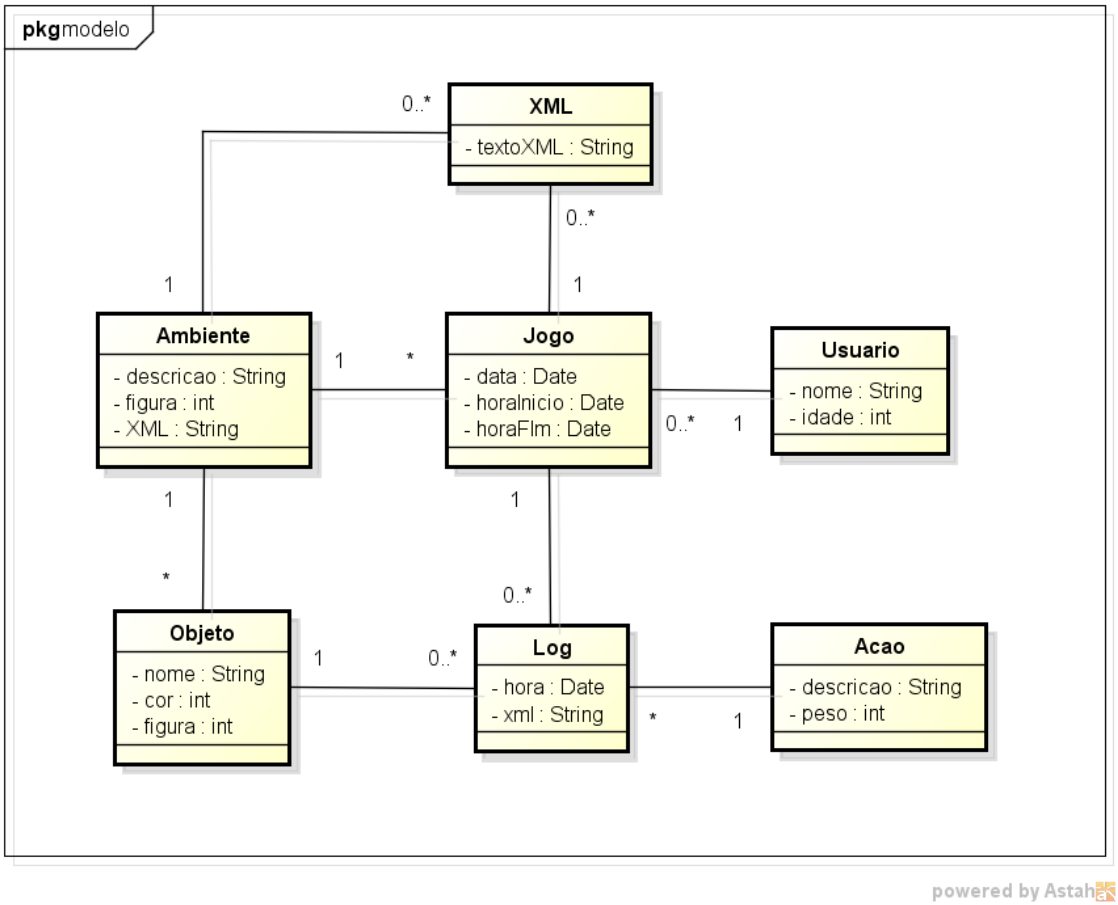


Figura 33. Diagrama de classes de entidades do modelo proposto

4. CONCLUSÕES

O objetivo geral deste trabalho foi propor um modelo para aprendizagem da linguagem de marcação XML. Para tanto foram atendidos os seguintes objetivos específicos:

● Ampliar os conhecimentos e práticas obtidas nas Unidades de ensino do Curso Superior em Gestão de Tecnologia da Informação - Algumas das unidades de ensino relacionadas com esta pesquisa são: PDS (projeto de desenvolvimento de sistemas), da qual obtivemos a metodologia para modelagem desse protótipo; Programação Web, que fundamenta os conceitos de linguagens de marcação; Programação orientada a objetos, que nos proveu modelo MVC; Introdução à informática, que nos propiciou os conceitos básicos; e Métodos e técnicas de pesquisa, que estabelece os fundamentos da pesquisa acadêmica.

● Pesquisar métodos e técnicas existentes para ensinar linguagens de computador para crianças - Conforme a pesquisa, jogos como os Serious Games, são amplamente utilizados em muitos dos métodos de ensino de linguagens de programação. Qualquer aplicativo hoje está fundamentado na abordagem de interface gráfica e intuitiva, daí a preferência por imagens ao texto para facilitar as interações homem X máquina.

● Explorar os conceitos relacionados às linguagens de marcação - O modelo proposto foi baseado nos conceitos aqui explorados. Partindo-se dos primórdios da web onde as páginas eram estáticas e com pouca interação, passando pela evolução de mecanismos como a Ajax, XML, web-semântica e as ontologias web.

● Usar uma metodologia para a modelagem do sistema - Exercício prático da metodologia proposta pela unidade de PDS aplicada a uma situação real e desafiante de aprendizagem, seja pelo público alvo, que não se restringe somente a pessoas alfabetizadas (ex. crianças em idade pré-escolar) e da compreensão de uma linguagem que se propõe a descrever nosso mundo e o mundo dos sistemas de informações, de forma que os próprios computadores passem a reconhecer a semântica dos dados sem a necessidade de sistemas intermediários como os programas de computador.

● Auxiliar no entendimento da linguagem de marcação XML de forma lúdica e em contextos de aprendizagem específicos - Não só para o projeto, mas pessoalmente este estudo permitiu que pudéssemos compreender melhor as aplicações que envolvem desde interoperabilidade de sistemas de informações até um possível advento de estruturas de dados que podem ser entendidas por sistemas de informações.

Como resultado geral obtemos um modelo inicial e passível de ser validado antes de sua implementação. Assim, atingimos nosso objetivo mais geral, embora saibamos que há, como veremos nas recomendações, ainda muito trabalho pela frente. Para efeito de conhecimento do pesquisador, muitas foram as descobertas: a web-semântica, linguagens de marcação, o processo de modelagem, uso da linguagem UML aplicado a um sistema, o processo de aprendizagem na criança e serious games.

5. RECOMENDAÇÃO PARA TRABALHOS FUTUROS

- Validar, com os possíveis usuários e especialistas, o modelo proposto, antes de sua implementação.
- Desenvolver a modelagem da arquitetura do sistema - definição dos artefatos, implantação, colaborações, padrões e frameworks, diagramas de artefatos e diagramas de implantação.
- Desenvolver um sistema para validar o modelo, testar com os públicos alvos propostos e verificar se os resultados são adequados.
- Analisar ontologias web possíveis de serem aplicadas a modelos desse tipo.
- Adaptar o modelo para o uso de Web Semântica, com possibilidade de integração com outros jogos e bases de dados.
- Modelar um ambiente compartilhado multi-usuário para que várias crianças possam jogar no mesmo ambiente com seus respectivos avatares.
- Testar e analisar os dados coletados para verificar se o modelo está sendo eficiente para aprendizagem da linguagem.
- Ampliar o modelo para funcionar com realidade virtual.

REFERÊNCIAS

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. UML – **Guia do Usuário**. Editora Campus, 2000.

BERNERS-LEE, T., LASSILA, Ora; HENDLER, James. **The semantic web**. Scientific America, maio, 2001. Disponível em: <https://www.scientificamerican.com/article/the-semantic-web/> Acesso em 24/março/2017

BISCOLI, 2005; p. 29, apud CORDAZZO e VIEIRA, **A brincadeira e suas implicações nos processos de aprendizagem e de desenvolvimento,** in Estudos e Pesquisas em Psicologia, jun/2007. Disponível em: <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1808-42812007000100009>, acesso em 03/05/2017)

BISCOLI, I. Â. **Atividade lúdica uma análise da produção acadêmica brasileira no período de 1995 a 2001.** 2005. Dissertação de Mestrado, Programa de Pós-Graduação em Educação, Universidade Federal de Santa Catarina, Florianópolis.

BLOG SONY XPERIA (Brasil) (Org.). **Uma em cada 5 crianças usa smartphones, *tablets* e aparelhos *mobile* todos os dias.** 2013. Disponível em: <http://www.blogsonyxperia.com.br/uma-em-cada-5-criancas-usa-smartphones-tablets-todos-os-dias/>. Acesso em: 20 abr. 2017.

CHALONBLANC, A., **Introdução a Jean Piaget**. Lisboa: Instituto Piage, 2001

CODE CLUB BRASIL (Brasil) (Org.), **Sobre o Projeto CCB**. Disponível em: <http://www.codeclubbrasil.org.br/sobre-o-projeto/> Acesso em 21/maio/2017.

CORDAZZO, S. T. D. e VIEIRA, M. L. **A brincadeira e suas implicações nos processos de aprendizagem e de desenvolvimento**. Florianópolis, mar/2007. Disponível em:

<http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1808-42812007000100009#mailfim> Acesso em 04/05/2017

COSTA, G. A., ANDRADE CHAGAS, A. A. e CHAGAS, E.H.P.B. **Benefícios da Tecnologia Para Crianças e Adolescentes**. Nov de 2016. Publicado em jan, 2017. Disponível em: <http://blog.smp.org.br/2017/01/19/tecnologia-criancas-adolescentes/> Acesso em jun/2017.

DANNY YAROSLAVSKI, ***Text is keeping kids from coding***. Disponível em: https://medium.com/@dannyyaroslavski/text-is-keeping-kids-from-coding-b70988dd7541. Acesso em: 28/ago/2017.

DECKER, S. et al. ***The semantic web: the roles of XML and RDF***. IEEE Expert, v. 15, n. 3. Oct. 2000.

DEVMEDIA. **Introdução ao padrão MVC**. Disponível em: <http://www.devmedia.com.br/introducao-ao-padrao-mvc/29308> Acesso em 19/10/2017.

EBC. **Crianças utilizam cada vez mais o *tablet* para acessar internet**. 2015. Disponível em: <http://www.ebc.com.br/infantil/para-pais/2015/08/criancas-utilizam-cada-vez-mais-o-tablet-para-acessar-internet/> Acesso em 10/mar/2017.

FANTIN, M. **Mídia-educação: conceitos, experiências, diálogos Brasil-Itália**. Florianópolis: Cidade Futura, 2006.

FANTIN, M. **Alfabetização midiática na escola**. Florianópolis, 2012. Disponível em: <http://alb.com.br/arquivo-morto/edicoes_anteriores/anais16/sem05pdf/sm05ss15_06.pdf/> Acesso em 11 abr, 2017.

FERNANDES, Felipe e MASSETTO, Francisco Isidro. **Pensamento computacional faz diferença na escola, diz professor**. Veja Entrevista, 29 de maio de 2017. Disponível em: <http://veja.abril.com.br/tveja/veja-

entrevista/pensamento-computacional-ajuda-nas-notas-escolares-diz-professor/> Acesso em: 30 maio, 2017.

FRIEDMANN, A. **O direito de brincar:** a brinquedoteca. 4ª ed. São Paulo: Abrinq, 1996.

FURGERI, Sérgio. **Ensino didático da linguagem XML**. São Paulo. Érica. 2001.

FURGERI, Sérgio. **O papel das linguagens de marcação para a Ciência da Informação**. TransInformação, Campinas, 18(3):225-239, set./dez., 2006.

GEE, James Paul. **Bons videogames e boa aprendizagem**. Perspectiva, v. 27, n. 1, 167-178, jan./jun. 2009. Disponível em: <https://periodicos.ufsc.br/index.php/perspectiva/article/view/2175-795X.2009v27n1p167/14515/> Acesso em: 07 abr. 2017.

GEE, James Paul. **What videogames have to teach us about learning and literacy**. New York: Palgrave/Macmillan, 2003.

GEE, James Paul. "**Video Games: do they have educational value?**," CQ Research, vol. 16, no. 10, pp. 939-959, 2006.

HOWARD, Jacqueline. **Kids under 9 spend more than 2 hours a day on screens, report shows**. CNN. 2017. Disponível em: <http://edition.cnn.com/2017/10/19/health/children-smartphone-tablet-use-report/index.html> Acesso em: 25/out/2017.

LEMES, David de Oliveira. **Serious Games. Jogos e educação**. 27/ago/2014. Disponível em: <http://www.abrelivros.org.br/home/index.php/bienal-2014/resumos-e-fotos/5647-primeiro-resumo> Acesso em: 25/10/2017.

LOPES, N., & OLIVEIRA, I. (2013). **Videojogos, Serious Games e Simuladores na Educação: usar, criar e modificar**. Educação, Formação & Tecnologias, 6(1), 04-20 [Online], disponível em:

<http://eft.educom.pt/index.php/eft/article/view/346/176> Acesso em 14 mar. 2017.

LUZ, Carla Sofia . Jornal de Notícias. **Pais usam televisor e tablet para entreterem os filhos.** 2017. Disponível em: <http://www.jn.pt/nacional/interior/pais-usam-televisor-e-tablet-para-entreterem-os-filhos-5634092.html/> Acesso em 10/mar/2017.

MODESTO, Fábio A. C. e SCAVACINI, Alexandre. **Utilização de games como apoio no processo ensino-aprendizagem**. São Paulo, out/2013. Disponível em: <https://www.researchgate.net/profile/Fabio_Alexandre_Caravieri_Modesto/publication/257872525_Utilizacao_de_Games_como_apoio_no_processo_ensino-aprendizagem/links/0c960525ff60f3548c000000.pdf/> Acesso em: 31/mar/2017.

MULLER, J. C. **Crianças na contemporaneidade: representações e usos das tecnologias móveis na educação infantil**. Dissertação de mestrado em educação. UFSC. Florianópolis, 2014.

NIELSEN. **Crianças e jovens da classe A utilizam mais o tablet do que adultos**. 2015. Disponível em: <http://www.nielsen.com/br/pt/press-room/2015/Criancas-e-jovens-da-classe-A-utilizam-mais-o-tablet-do-que-adultos-aponta-estudo-da-Nielsen-Ibope.html/> Acesso em 10/mar/2017.

SANTOS, P. P. P. O.; PAULINO, S. F.; CRUZ, C. F.; FLORIANO, J. I. **Desenhos animados e educação na pré-escola: um estudo de caso dos principais canais infantis de TV no Brasil**. III CONEDU. 2015. Disponível em: <http://www.editorarealize.com.br/revistas/conedu/trabalhos/TRABALHO_EV056_MD1_SA19_ID11585_18082016173408.pdf> Acesso em 05/06/2017.

SOUZA, Gilmar Carvalho. **Página com as Unidades de Cursos do IFSC ministradas por esse professor**. Disponível em: gilmarcarvalho.com.br. Acesso em novembro de 2017.

SOUZA, Renato Rocha e ALVARENGA, Lídia . **A Web Semântica e suas contribuições para a ciência da informação**. Ci. Inf., Brasília, v. 33, n. 1, p. 132-141, jan./abr. 2004. Disponível em: <http://www.scielo.br/pdf/ci/v33n1/v33n1a16/> Acesso em 24/mar/2017.

SILVERSTONE, R. **Entrevista: Inventar o quinto poder**, por Ubiratan Muarrek. Ed. 227. Carta Capital, 2003. Disponível em: <http://www.usp.br/nce/wcp/arq/textos/65.pdf/> Acesso em: 07 abr. 2017.

TECMUNDO (Org.). **Até que ponto a tecnologia faz mal na infância?** 2012. Disponível em: <https://www.tecmundo.com.br/estilo-de-vida/32723-ate-que-ponto-a-tecnologia-faz-mal-na-infancia-.htm/>. Acesso em: 10 mar. 2017.

VEJA.COM (Brasil) (Org.) **Entrevista: Pensamento computacional faz diferença na escola, diz professor.** 2017. Disponível em: <http://veja.abril.com.br/tveja/veja-entrevista/pensamento-computacional-ajuda-nas-notas-escolares-diz-professor/>. Acesso em: 29 maio 2017.

VYGOTSKY, L. S. **A formação social da mente:** o desenvolvimento dos processos psicológicos superiores. 4ª ed. São Paulo: Martins Fontes, 1991.

W3C. **Padrões da Web Semântica**. Disponível em: <http://www.w3c.br/Padroes/WebSemantica/> Acesso em 22/mar/2017.

W3Schools. **Curso de HTML Básico**. Disponível em: https://www.w3schools.com/html/html_basic.asp. Acesso em 02/jun/2017.

WINNICOTT, D.W. **O Brincar e a Realidade**. Rio de Janeiro. Editora: IMAGO, 1975. Disponível em: <http://docslide.com.br/documents/winnicott-dw-o-brincar-e-a-realidadepdf-56264daf1b074.html> Acesso em 16/jun/2017.

APÊNDICE F – Declaração de finalização de trabalho de curso

INSTITUTO FEDERAL
SANTA CATARINA

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS

DECLARAÇÃO DE FINALIZAÇÃO DE TRABALHO DE CURSO

Declaro que o(a) estudante Gabriela Andrea Sorel, matrícula nº 151005249-6, do Curso Superior de Gestão da Tecnologia da Informação, defendeu o trabalho intitulado Modelo de Aprendizagem Lúdica de XML Baseado em um Jogo de Computador, o qual está apto a fazer parte do banco de dados da Biblioteca Hercílio Luz do Instituto Federal de Santa Catarina, Câmpus Florianópolis.

Florianópolis, 05 de fevereiro de 2018.

Prof. Orientador do TCC

Prof. Julio Cesar C. Ribas
Portaria: DOU Nº 566 - 24/02/2017
Coord. CST - CTI
IFSC - Câmpus Florianópolis