

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**GUILHERME COSTA BERGHAUSER**

**PRODUÇÃO DE REA PARA PREVENÇÃO DA VIOLÊNCIA SEXUAL  
INFANTIL**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PATO BRANCO  
2015**

**GUILHERME COSTA BERGHAUSER**

**PRODUÇÃO DE REA PARA PREVENÇÃO DA VIOLÊNCIA SEXUAL  
INFANTIL**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Ms. Vinicius Pegorini

Coorientadora: Prof. Ms. Ana Carla Vagliati

**PATO BRANCO  
2015**

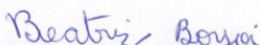
ATA Nº: 006

DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO DO ALUNO  
GUILHERME COSTA BERGHAUSER.

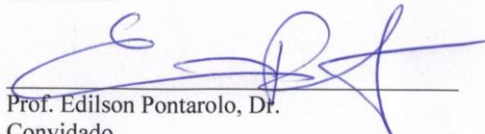
Às 13:35 do dia 26 de novembro de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Vinicius Pegorini (orientador), Beatriz Terezinha Borsoi (convidada) e Edilson Pontarolo (convidado), para avaliar o Trabalho de Conclusão de Curso do aluno Guilherme Costa Berghauser, matrícula 1494104, sob o título **Produção de REA para prevenção da violência sexual infantil**; como requisito final para a conclusão da disciplina Trabalho de Conclusão de Curso 2 do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 14:35 hrs foi encerrada a sessão.



Prof. Vinicius Pegorini, M.Sc.  
Orientador



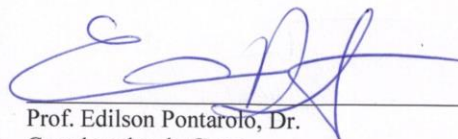
Profa. Beatriz Terezinha Borsoi, Dr.  
Convidada



Prof. Edilson Pontarolo, Dr.  
Convidado



Profa. Soelaine Rodrigues Ascari, M.Sc.  
Coordenador do Trabalho de Conclusão de Curso



Prof. Edilson Pontarolo, Dr.  
Coordenador do Curso

## RESUMO

BERGHAUSER, Guilherme Costa. Produção de recurso educacional aberto para prevenção da violência sexual infantil. 2015. 50 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

A escola, instituição primordial de acesso ao conhecimento, também é afetada pela violência sexual ao ter em sala de aula alunos vítimas dessa forma de violência. Enfatiza-se o papel da escola de educar e ressalta-se o essencial trabalho de prevenção e de identificação de casos de violência sexual. A partir deste desafio surge o software educativo que será apresentado neste estudo com vistas à melhoria da qualidade educacional. O software se constitui num editor de histórias em quadrinhos (HQs) para ser utilizado em sala de aula como uma ferramenta de expressão do aluno vítima de violência sexual (identificação) e, de criação de histórias, com o intuito de prevenção desta violência.

**Palavras-chave:** JavaScript. MEAN. Educação Sexual. Violência Sexual. REA.

## ABSTRACT

BERGHAUSER, Guilherme Costa. Open educational resource production for prevention of child sexual violence. 2015. 50 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

The school, primary institution of access to knowledge is also affected by sexual violence to have in class pupils victims of such violence. We emphasize the role of schools to educate and its essential work of prevention and identification of cases of sexual violence. From this challenge arises educational software that we will be presenting in this study aiming to improving educational quality. The software is an editor of comic books for use in the classroom as a tool of expression of the student victims of sexual violence allowing identification, and creating stories, with the aim of preventing such violence.

**Keywords:** JavaScript. MEAN. Sexual Education. Violence Sexual. OER.

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 - Página Inicial do Pixton.....                    | 24 |
| Figura 2 - GoAnimate, Tela de Inserção/Edição de Cenas..... | 24 |
| Figura 3 - Livro Digital do ToonDoo .....                   | 25 |
| Figura 4 - Diagrama de Caso de Uso.....                     | 32 |
| Figura 5 - Tela Criar Quadrinho .....                       | 33 |
| Figura 6 - Tela Edição de Personagem .....                  | 34 |
| Figura 7 - Edição de Balão de Fala.....                     | 35 |
| Figura 8 - Meus Quadrinhos .....                            | 36 |
| Figura 9 - Tela Modo de Leitura .....                       | 36 |
| Figura 10 - Tela Inserir Tarefa .....                       | 37 |
| Figura 11 - Tela Minhas Tarefas .....                       | 38 |
| Figura 12 - Configuração.....                               | 38 |

## LISTA DE QUADROS

|   |    |
|---|----|
| Quadro 1 - Descrição das atribuições das licenças Creative Commons..... | 20 |
| Quadro 2 - Lista de materiais utilizados no sistema.....                | 26 |
| Quadro 3 - Lista de requisitos funcionais.....                          | 32 |
| Quadro 4 - Lista de requisitos não funcionais. ....                     | 32 |

## LISTA DE SIGLAS

|        |  |
|--------|--|
| ACID   | Atomicidade, Consistência, Isolamento e Durabilidade             |
| API    | <i>Application Programming Interface</i>                         |
| CSS    | <i>Cascading Style Sheets</i>                                    |
| ECA    | Estatuto da Criança e do Adolescente                             |
| HQ     | Histórias em Quadrinhos  |
| HTML   | <i>HyperText Markup Language</i>                                 |
| HTTP   | <i>HyperText Transfer Protocol</i>                               |
| JS     | JavaScript   |
| JSON   | <i>JavaScript Object Notation</i>                                |
| LDA    | Lei de Direitos Autorais   |
| MEAN   | <i>MongoDB, Express, AngularJS, Node.js</i>                      |
| MVC    | <i>Model-View-Controller</i>                                     |
| MVR    | <i>Model-View-Routes</i>   |
| NoSQL  | <i>Not Only SQL</i>  |
| OER    | <i>Open Educational Resource</i>                                 |
| REA    | Recursos Educacionais Abertos                                    |
| SQL    | <i>Structured Query Language</i>                                 |
| TDD    | <i>Test Driven Development</i>                                   |
| UNESCO | Organização das Nações Unidas para a Educação, Ciência e Cultura |
| XP     | <i>Extreme Programming</i>                                       |
| YAGNI  | <i>You aren't gonna need it</i>                                  |



## SUMÁRIO

|   |           |
|---|-----------|
| <b>1 INTRODUÇÃO</b> .....               | <b>9</b>  |
| 1.1 CONSIDERAÇÕES INICIAIS .....        | 9         |
| 1.2 OBJETIVOS .....                     | 10        |
| 1.2.1 Objetivo Geral .....              | 10        |
| 1.2.2 Objetivos Específicos .....       | 11        |
| 1.3 JUSTIFICATIVA .....                 | 11        |
| 1.4 ESTRUTURA DO TRABALHO .....         | 12        |
| <b>2 REFERENCIAL TEÓRICO</b> .....      | <b>13</b> |
| 2.1 VIOLÊNCIA E VIOLÊNCIA SEXUAL .....  | 13        |
| 2.2 EDUCAÇÃO SEXUAL EMANCIPATÓRIA ..... | 14        |
| 2.3 RECURSOS EDUCACIONAIS ABERTOS ..... | 17        |
| 2.4 AGILE .....                         | 20        |
| 2.4.1 Manifesto Ágil .....              | 20        |
| 2.4.2 Scrum .....                       | 21        |
| 2.4.3 Extreme Programming .....         | 22        |
| 2.5 HISTÓRIAS EM QUADRINHOS .....       | 23        |
| 2.5.1 Pixton .....                      | 23        |
| 2.5.2 GoAnimate .....                   | 24        |
| 2.5.3 Toondoo .....                     | 25        |
| <b>3 MATERIAIS E MÉTODO</b> .....       | <b>26</b> |
| 3.1 MATERIAIS .....                     | 26        |
| 3.2 JAVASCRIPT .....                    | 26        |
| 3.3 MEAN .....                          | 27        |
| 3.3.1 Node.JS .....                     | 27        |
| 3.3.2 Express .....                     | 28        |
| 3.3.3 AngularJS .....                   | 29        |
| 3.3.4 MongoDB .....                     | 29        |
| 3.3 MÉTODO .....                        | 30        |
| <b>4 RESULTADOS</b> .....               | <b>31</b> |
| 4.1 ESCOPO DO SISTEMA .....             | 31        |
| 4.2 MODELAGEM DO SISTEMA .....          | 31        |
| 4.3 APRESENTAÇÃO DO SISTEMA .....       | 33        |
| 4.4 IMPLEMENTAÇÃO DO SISTEMA .....      | 38        |
| <b>5 CONCLUSÃO</b> .....                | <b>45</b> |
| REFERÊNCIAS .....                       | 46        |

## 1 INTRODUÇÃO

Este capítulo apresenta a introdução do trabalho que é composta pelas considerações iniciais com o escopo e contexto do trabalho, os objetivos e a justificativa. O capítulo é finalizado com a apresentação dos capítulos subsequentes.

### 1.1 CONSIDERAÇÕES INICIAIS

A violência sexual contra crianças tem ganhado ênfase na sociedade, especialmente por meio da mídia, que ainda trata de maneira tímida e superficial tal questão, mas que traz à tona informações que outrora se restringiam aos estudiosos da área. Parte-se da compreensão de mídia contida no dicionário, que é entendida como meios de comunicação de massa, são eles: a televisão, o rádio, a internet, o telefone, o jornal impresso, o cinema, etc.

Mesmo ganhando visibilidade social, ao mesmo tempo em que a população sente desprezo por atos de violência sexual contra crianças, silencia frente a tal fenômeno (Vagliati, 2014). Assim, destaca-se a importância de voltar o olhar para a violência sexual contra crianças e buscar dar-lhe visibilidade.

Para Vagliati (2014), junto à denúncia e identificação, se faz necessário o papel da escola no enfrentamento à violência sexual por meio da educação. A autora acredita que, o professor ao educar crianças e adolescentes sobre sexualidade e violência sexual poderá ajudar na concretização de uma das formas de enfrentamento a essa violência. Nesse sentido, Gagliotto e Lembeck (2011) defendem que,

[...] a Educação Sexual Emancipatória pressupõe um trabalho educativo comprometido em promover a autonomia do educando, buscando superar padrões de comportamentos hierarquizados e estereotipados, superando preconceitos e tabus através da compreensão dos aspectos sócio-histórico-políticos que influenciaram na sua construção. Partindo dessa compreensão, pode-se investir na reconstrução de novos padrões para a vivência de uma sexualidade plena, que resgate a dimensão humana, o respeito e a busca do prazer sem medo e sem culpas. (p. 16).

Não se deve ser ingênuo e defender que a escola sozinha dará conta do enfrentamento à violência sexual. Assim, considera-se necessária a integração de todas as demais instituições que

lidam com crianças e adolescentes para, num trabalho cooperativo e articulado, buscarem a prevenção e o atendimento às situações de violência e/ou vulnerabilidade.

Entre as instituições que compõem a rede de proteção à criança, destaca-se a escola e seus professores. No entanto, sabe-se que essa entidade apresenta limites para promover a prevenção da violência sexual. Vagliati (2014), ao pesquisar as instituições escolares e seus professores no município de Francisco Beltrão – PR, pode constatar que apenas 20% dos professores entrevistados por ela possuíam formação na área da sexualidade e da violência sexual.

Defende-se nesse contexto a utilização pelos professores de recursos educacionais digitais, para que assim, ao educarem sobre a sexualidade estejam promovendo a prevenção da violência sexual. Gagliotto (2009) e Vagliati (2014) destacam a mídia como um canal possível no debate e na difusão de informações relevantes sobre a violência sexual. Parte-se da compreensão das autoras e defende-se o potencial da mídia, no sentido de ampliar a rede de proteção, promover discussões e se tornar efetiva na prevenção a essa violência. É lamentável notar que ainda hoje, na maioria das vezes, a mídia tende a trazer informações distorcidas sobre a vítima, culpabilizando-a pela violência sofrida ou mesmo naturalizando esta violência.

Nesse contexto, o presente estudo tem como finalidade desenvolver um editor de histórias em quadrinhos (HQ) para trabalhar com a educação sexual e com a identificação da violência sexual de crianças e de adolescentes. Como recurso, a HQ possibilita a edição dos personagens e a escolha de cenas, tendo como finalidade se adequar a realidade das crianças e dos adolescentes. Assim, após escolher um, dentre os diversos cenários, é possível alterar os personagens que irão compô-lo, como, por exemplo, escolher entre: adulto, adolescente, criança e idoso, alterar as expressões faciais, o tom de pele, o cabelo, as roupas, etc, e também acrescentar objetos.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Desenvolver uma aplicação web sobre educação sexual visando à prevenção da violência sexual infantil.

### 1.2.2 Objetivos Específicos

- Realizar buscas na internet sobre a existência de aplicativos de histórias em quadrinhos;
- Entrevista com profissional da área de educação sexual, para levantar dados a respeito de suas necessidades no trabalho com as crianças e adolescentes;

### 1.3 JUSTIFICATIVA

A motivação inicial que impulsionou pesquisar o fenômeno da violência sexual advém dos relatos dos professores entrevistados por Vagliati (2014). Em sua pesquisa, a autora busca entender qual a compreensão dos professores sobre sexualidade e violência sexual. Fica evidente, nas falas dos professores, tamanha dificuldade em tratar estes temas em sala de aula; relatam que são de uma geração que não se falava sobre sexo, mas vivem em uma sociedade do século XXI, que Chauí (1984) categoriza como a “sociedade tagarela”, que fala sobre sexo, mas impõe e gera repressão.

A sexualidade não é bem trabalhada; as pessoas têm muito medo, em especial nós professores. [...] Os professores precisam ter uma melhor formação para lidar com a sexualidade. Os professores de hoje são da geração que não se falava de sexo. (CHAUÍ, 1984, p. 104)

Segundo a pesquisa de Cunningham e Sas (1995), na qual buscaram entender o papel e as ações que as escolas têm frente à violência sexual infantil, apontam que em 44% dos casos de violência sexual os professores eram a primeira pessoa a saber, e em 52% o primeiro adulto a saber. Esses dados demonstram o papel fundamental que o professor tem para quebrar a barreira do silêncio e a denúncia dos casos.

Figueiró (2006) e Vagliati (2014) defendem que a educação sexual deve ser emancipatória, pautada no respeito e no conhecimento do ser humano, que possibilite o indivíduo educar-se sexualmente, ou seja, promover a qualidade e afetividade nas relações sociais e sexuais, que permita rever os papéis sexuais, os tabus, os preconceitos e as discriminações.

Vagliati (2014) listou os materiais educativos sobre violência sexual de cunho governamental disponibilizados na internet, totalizando dezenove (19) materiais educativos, que podem ser utilizados pelos professores, sendo eles:

[...] seis (6) voltados, especificamente, para a prevenção, combate, identificação, enfrentamento e proteção da violência sexual, na área da educação. Quatro (4) se referem à área jurídica e dois (2) à área da saúde. Sete (7) dos materiais são para acesso do grande público. Destacamos que, embora alguns materiais estejam direcionados para uma área específica de atuação, a violência sexual é tratada de forma ampla, assim os materiais podem ser utilizados por profissionais das mais diversificadas áreas de formação. (VAGLIATI, 2014, p. 130).

Ainda, segundo a autora, em sua maioria, os professores não têm acesso a materiais sobre a violência sexual e temem realizar busca na internet por materiais aleatórios por não saberem quais realmente podem ser utilizados de forma educativa.

Esse contexto permite levantar alguns questionamentos: Como a tecnologia pode contribuir com a educação sexual com vistas a prevenção da violência sexual? Quais materiais didáticos existem nessa temática? Quais são as abordagens desses materiais sobre a violência sexual? Qual a importância e a eficiência de materiais didáticos interativos para o ensino da educação sexual? Os professores têm acesso a esse tipo de material?

A partir de tais considerações, o presente estudo pretende criar um material didático interativo sobre educação sexual para prevenção da violência sexual, e buscar alternativas para professores trabalharem em sala de aula com seus alunos e enfrentarem o grave problema da violência sexual.

#### 1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos. Este é o primeiro e apresenta as considerações iniciais, o objetivo e a justificativa do trabalho. O Capítulo 2 apresenta o referencial teórico que se refere ao desenvolvimento de aplicações web. No Capítulo 3 estão os materiais e o método utilizados para a modelagem e a implementação do sistema. Os resultados da realização deste trabalho são apresentados no Capítulo 4. Por fim está a conclusão, seguida das referências bibliográficas.

## 2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho, no qual são apresentados os conceitos de violência e violência sexual, interligando essa temática com a educação emancipatória, referenciando os recursos educacionais abertos (REA) e descrevendo os recursos.

É discorrido sobre a linguagem de desenvolvimento JavaScript (JS), bem como, é descrito sobre as tecnologias MongoDB, Node.js, AngularJs e Express, que a partir delas surge o MEAN.

### 2.1 VIOLÊNCIA E VIOLÊNCIA SEXUAL

Etimologicamente a palavra violência tem origem do latim *vis* (violência), conforme o tempo ela derivou as palavras vida, do latim *vitae* e todas suas variações como: vitalidade, vigor, vitamina; como também violência e suas variações: violar, violação. Zimmerman (2012) faz um paralelo a essa definição, segundo ele a transição da palavra vigor sadio para o de violência é o mesmo que quando ocorre uma agressividade sadia para uma agressividade destrutiva.

Vagliati (2014) ao tratar sobre o conceito de violência contra crianças e adolescentes afirma que os estudiosos da área têm como base a teoria do poder para explicar esse tipo de violência, é destacado que nem todo poder resulta em violência, porém, todo poder onde há sobreposição de um sobre o outro, exercida intencionalmente, visando obter vantagens sendo elas sexuais, de dominação e/ou lucro passa a ter uma relação violenta que nega os direitos do dominado e desestrutura a sua identidade.

Vagliati (2014) com base nos estudos de Zimmerman (2012) discorre que:

O conceito de poder é amplo. Em linhas gerais, pode-se defini-lo como sendo a capacidade ou possibilidade de agir e produzir efeitos desejados sobre outros indivíduos ou grupos humanos. Implica em exercer poder e, reciprocamente, poder efetivá-lo. Tem raiz etimológica no étimo latino *pot* e significa *aquele que pode, que é capaz*, com derivação como nos verbos *possum* e *possidere* que se forma de *pot-potis* (poder) + *sedere* (sentar), ou seja, *aquele que senta no trono e possui todo o poder*; e nas palavras *potência, impotência, onipotência, prepotência*. (Vagliati, 2014, p. 29-30)

Arendt (1994) procura analisar politicamente a relação entre o poder e a violência, para a autora o poder se caracteriza por ser fruto de uma ação coletiva, ou seja, o poder pertence a um grupo e só permanece nele apenas quando o mesmo permanece unido. Eles ainda são capazes de eleger uma pessoa que “estará no poder” para representar a todos do grupo. A violência entra em foco, devido ao enfraquecimento desse poder, a autora descreve esse fato como, a impossibilidade de um fazer conjunto.

Também Chauí (1984) atenta seus estudos sobre a violência e busca defini-la. A autora alerta que cada sociedade e cultura vê esse fenômeno de forma diferente, contudo alguns aspectos são percebidos da mesma forma nessas culturas e sociedades distintas, assim pontua:

[...] a violência é percebida como exercício da força física e da coação psíquica para obrigar alguém a fazer alguma coisa contrária a si, contrária aos seus interesses e desejos, contrária ao seu corpo e à sua consciência, causando-lhe danos profundos e irreparáveis, como a morte, a loucura, a auto-agressão ou a agressão aos outros.(p. 336)

Pode-se perceber que a violência se manifesta de múltiplas e diferentes formas, uma delas é a violência sexual. A discussão sobre essa forma de violência tomou força nos anos 90, devido a dois grandes marcos políticos nos direitos humanos, com a promulgação da Carta Constitucional de 1988 e em 1990 pela aprovação do Estatuto da Criança e do Adolescente (ECA), que estabelece a obrigatoriedade da sociedade na defesa das crianças e dos adolescentes. Vagliati (2014) acrescenta “Anteriormente a esse período, a criança e o adolescente não ocupavam lugar de preocupação na legislação e na sociedade civil.”, ou seja, essas leis foram de grande avanço nos direitos humanos e nas discussões sobre os direitos das crianças e adolescentes, tornando o estado conhecedor desse problema social e, chamando-o para o papel de protetor, ao defender e garantir os direitos das crianças e dos adolescentes.

Segundo Faleiros e Faleiros (2008) entende-se que a violência sexual se constitui:

[...] através do contato físico, ou seja, por meio de carícias não desejadas, penetração oral, anal ou vaginal, com o pênis ou objetos, masturbação forçada, dentre outros; e sem contato físico, por exposição obrigatória a material pornográfico, exibicionismo, uso de linguagem erotizada em situação inadequada. (p. 39).

Azevedo e Guerra (2007) e Santos (2011) se preocupam em caracterizar esse fenômeno, dividindo-os em três grupos, sendo eles: (a) não envolvendo contato físico; (b) envolvendo contato físico; (c) envolvendo o uso da força; a justificativa para tal divisão é alertar que muitas vezes a violência sexual não deixam marcas visíveis no corpo, mas o estrago gerado na vida da vítima é incalculável, Santos (2011, p. 55) completa “o estrago que podem causar na história de vida desta criança ou adolescente, não podemos nem ousar dizer que imaginamos qual seja”.

## 2.2 EDUCAÇÃO SEXUAL EMANCIPATÓRIA

Pogrebinschi (2004) e Adorno (2006), em seus estudos, procuram compreender o conceito de emancipação, por muitas vezes ambos demonstram preocupações quanto ao conceito, segundo

eles o termo “emancipação” utilizado por muitos/as autores/as é relacionada apenas ao significado de liberdade e/ou igualdade. Para Adorno (2006), o termo precisa ser melhor investigado:

[...] a idéia de emancipação, como parece inevitável com conceitos deste tipo, é ela própria ainda demasiado abstrata, além de encontrar-se relacionada a uma dialética. Esta precisa ser inserida no pensamento e também na prática educacional. (p. 143).

Diante dessa problemática se faz necessário conceituar “emancipação”, em sua formulação latina original *emacipatio*, deriva de “manu capare”, que é um ato jurídico da Roma antiga, utilizadas pelas *paterfamilias* da República Romana para libertar seus filhos do pátrio poder (POGREBINSCHI, 2004), ou seja, seus filhos só eram “emancipados” após completarem três (3) vendas sucessivas, chamadas de *mancipationes*, além de completar muitas outras missões estipuladas pelo pai (SILVA, 2005).

Depois desse período somente com os iluministas o termo voltou a estar em voga, “o período iluminista exaltava a liberdade de pensamento, os valores morais, éticos e religiosos somados a emancipação política [...] almejavam alcançar os objetivos maiores que seriam a liberdade, a autonomia e a emancipação.” (DECKER, 2010, p. 30). Para Kant o termo emancipação está ligada a pessoa atingir certa idade e sair da minoridade. É o “esforço do sujeito se desvencilhar da dependência do outro e tornar-se esclarecido.” (DECKER, 2010, p. 30). É importante destacar que tanto em Roma como na corrente iluminista o termo “emancipação” está relacionado ao termo jurídico (leis), onde esse contexto só se alterou na corrente Marxista.

Pogrebinschi (2004) contextualiza essa troca de conceitos resultantes da corrente Marxista:

A superação do Estado, o *aufhebung*, constitui o momento máximo da mancipação – e não apenas da emancipação política, que é apenas uma etapa de algo maior, a emancipação humana. (...) em Marx a emancipação é sempre um processo auto-reflexivo, auto-executável. Em outras palavras, a emancipação é sempre auto-emancipação, mesmo que estejamos falando de sujeitos coletivos, como ‘os judeus’, ‘os cidadãos’ ou ‘a humanidade’. (p. 5).

Nesse momento histórico surge o conceito de “*emancipação humana geral*”, parafraseando Decker (2010, p. 34) o novo conceito é “a mais elevada expressão das potencialidades humanas”, Marx (2007) explana que:

somente quando o homem individual real recuperar em si o cidadão abstrato e se converter, como homem individual, em ser genérico, em seu trabalho individual e em suas relações individuais, somente quando o homem tenha reconhecido e organizado suas “forças próprias” como as forças sociais e quando, portanto já não separa de si a força social sob a forma de força política, somente então processa a emancipação humana. (p. 42)



É nesse contexto que a educação emancipatória se faz presente e necessária. Adorno (2006) afirma que a educação deve contemplar a emancipação e não somente um ser isolado, mas sim um ser social. Uma educação que se distingue da educação severa, autoritária e que não prioriza a conscientização.

Assim, uma educação emancipatória é aquela que prepara o sujeito para as adversidades do mundo, que o sujeito tenha condições de formular suas próprias ideias e indagações, tenha capacidade de autorreflexão e assim evite o desencadeamento de novas tragédias. É “uma educação voltada à conscientização e à racionalidade, uma educação como possibilidade de uma proposta emancipadora, uma emancipação.” (DECKER, 2010, p. 36).

No campo da sexualidade, uma abordagem emancipatória busca a liberdade do sujeito de qualquer repressão e autoritarismo da sociedade, Nunes (1996) foi quem propôs o termo “emancipatório” para sexualidade. Segundo Marques (2002, p.15) “o termo ‘emancipatório’ utilizado por este autor supõe uma profunda reflexão sobre a sexualidade, de modo a elucidar contradições históricas, discutir suas bases antropológicas, investigar suas matrizes sociológicas e identificar suas configurações políticas.”.

Nunes (1996) discorre que:

A sexualidade numa visão emancipatória supõe também normas, limites como marcos de sujeitos plenos e não sanções, pecados e medos. [...] Não há sociedade sem a normatização da sexualidade. [...] A normatização recusa a anomia e a heteronomia e propõe a autonomia, propõe o equilibrado conceito de compreender as contradições e superá-las dinâmica, arbitrária e dramaticamente no cotidiano, no dia-a-dia, nas condições reais do nosso viver. (p. 228 – 229).

É fundamental encarar a sexualidade como algo natural, inerente ao ser humano e, para que os sujeitos possam ser plenos, se faz fundamental que a temática seja trabalhada e discutida sem preconceitos. Landini (2011) coloca em pauta a discussão da violência sexual e corrobora com os autores citados anteriormente, segundo ela “pensar possíveis soluções para o problema da violência sexual implica pensar crianças e adolescentes como sujeitos ativos – proteger da violência e do abuso não significa isolá-los do mundo (real e virtual), mas prepará-los para lidar com essas situações.” (p. 97-98).

## 2.3 RECURSOS EDUCACIONAIS ABERTOS

Em paralelo a discussão de emancipação, emerge, no campo da tecnologia, o conceito de Recursos Educacionais Abertos, resultante da mudança de paradigma causada pela WEB 2.0. Segundo Santarosa *et al.* (2013) a WEB 1.0, chamada de primeira geração, caracterizada por ser uma grande biblioteca de informações, o usuário passou a ter acesso a uma grande quantidade de conteúdo, porém não podia modificá-las.

Com o passar dos tempos o perfil dos usuários foi se alterando, devido à ampliação das tecnologias de produção/geração de conhecimento, surge então a “Era do Usuário”, denominado por Santarosa *et al.* (2013) de geração interativa, os autores a conceituam:

Geração Interativa, produzida sob o conceito da inteligência coletiva e explicitada pelas múltiplas possibilidades de partilha e cooperação. Os atuais sistemas web projetam a personalização da navegação na Web, com programas que percebem especificidades e preferências do usuário, otimizando a capacidade de organizar e analisar informação. (p. 2).

Morais *et al.* (2013) discorrem sobre o cenário da WEB 2.0:

A Web 2.0 evoluiu para uma plataforma com potencialidades quer para a apresentação de informação, quer para a participação na criação de informação, podendo-se considerar como um conjunto de tecnologias e aplicações de software que permitem interagir, colaborar, criar e partilhar informações. (p. 54)

No contexto da WEB 2.0 surge a educação aberta que segundo Santarosa *et al.* (2013) emergiu “dos movimentos que questionavam a lógica disciplinar do processo educacional, [...] na autoridade do professor como detentor do conhecimento e no cerceamento do saber por uma rígida estrutura curricular” (p.4). Para Santos (2012) a educação aberta possui um conjunto de princípios que os estrutura, sendo eles:

1. A liberdade de escolha do aprendiz para estabelecer o tempo e o espaço para o aprender;
2. O respeito ao ritmo e ao estilo de aprendizagem dos estudantes;
3. A necessidade de desenvolver a autoinstrução;
4. A eliminação de requisitos prévios para acesso aos espaços educativos;
5. A possibilidade de acesso a alunos com deficiência ou em desvantagem social;
6. A disponibilização de recursos educacionais abertos, livres e acessíveis.

A partir desse momento a educação não se restringe aos limites impostos pelo desenho organizacional da escola tradicional, com traços ditatoriais e moldes industriais. Seu maior

diferencial está na possibilidade de alunos e professores poderem operar os materiais didáticos, pois dá aos autores tecnologias que os possibilitam criar, moldar e desenvolver conhecimentos de forma conjunta, concomitante a isso “institui um conjunto de novos saberes e competências para promoção de tempos/espços diversificados de aprendizagem formal e não-formal.” (SANTAROSA *et al.*, 2013, p. 4).

O ensino não-formal está empoderando usuários a ter acesso a muitos conteúdos disponibilizados pelos repositórios de áudio, vídeo, apresentações, textos, entre outros, segundo Santarosa et al. (2013) é um “processo de empoderamento do usuário”. Perskind e Zimmerman (1995), definem empoderamento como:

[...] à capacidade dos indivíduos e grupos poderem inferir e propor ações alternativas em múltiplas esferas. Para esses pesquisadores, esse conceito aproxima-se da noção de **autonomia**, uma vez que aponta para o poder dado aos indivíduos para interferir em temas que afetam a vida social, como também da perspectiva **emancipatória**, pois, empoderar é o processo pelo qual indivíduos, com unidades e organizações obteriam recursos que lhe permitam ter voz, visibilidade, influência, capacidade de ação e de tomada de decisão. (p. 5).

A definição de REA são diversas, porém existe um consenso entre os estudiosos da área, a proposta mais aceita é a criada pela Organização das Nações Unidas para a Educação, Ciência e Cultura (UNESCO) em 2002, reforçada ainda pela Declaração de Paris sobre REA, um documento emblemático sobre recursos educacionais abertos, segundo a UNESCO (2012):

REA são materiais de ensino, aprendizagem e investigação em quaisquer suportes, digitais ou outros, que se situem no domínio público ou que tenham sido divulgados sob licença aberta que permite acesso, uso, adaptação e redistribuição gratuitos por terceiros, mediante nenhuma restrição ou poucas restrições. O licenciamento aberto é construído no âmbito da estrutura existente dos direitos de propriedade intelectual, tais como se encontram definidos por convenções internacionais pertinentes, e respeita a autoria da obra. (Sem Página)




Muitos autores que tratam a caracterização do REA frisam sobre a construção colaborativa desses recursos, outros autores trabalham afirmando que em todo REA é preciso garantir os chamados “Quatro (4) Rs”, que são: **reúso**(1) reutilização desse recurso por qualquer pessoa, **revisão**(2) revisão (acesso ao código) por outras pessoas, **remix**(3) modificar ou reconstruir esses recursos para suprir a necessidade local do sujeito, **redistribuição**(4) permitir que esse recurso possa ser redistribuído e utilizado por outras pessoas (EMPIDOLA *et al.*, 2014).




Ao adentrar nessa discussão, perpassa-se a área de “direitos autorais”. No Brasil os direitos morais do autor são inalienáveis, ou seja, é defendido por lei, sendo ela a nº 9.610, de Direitos Autorais (LDA), de 19 de fevereiro de 1998 em seu art. 24, diz quais os direitos que o autor possui:

- I. O de reivindicar, a qualquer tempo, a autoria da obra;
- II. O de ter seu nome, pseudônimo ou sinal convencional indicado ou anunciado, como sendo o do autor, na utilização de sua obra;
- III. O de conservar a obra inédita;
- IV. O de assegurar a integridade da obra, opondo-se a quaisquer modificações ou à prática de atos que, de qualquer forma, possam prejudicá-la ou atingi-lo, como autor, em sua reputação ou honra;
- V. O de modificar a obra, antes ou depois de utilizada (BRASIL, 1998).

É importante frisar que, por mais que as obras estejam disponíveis para outras pessoas na internet, elas continuam a seguir as mesmas leis do “mundo real”, ao concluir isso, é imprescindível garantir que esses materiais proporcionem às pessoas a possibilidade de utilizar os Quatro (4) R. Para isso surgem as licenças chamadas *Creative Commons*, uma organização sem fins lucrativos, que busca criar padrões para abertura das obras. O Copyright é a lei oposta à tratada aqui, ela busca criar padrões para restringir o compartilhamento.

As licenças *Creative Commons*, são divididas pelas suas atribuições de liberdades ao usuário, (EMPIDOLA; PEREIRA; ALVES, 2014). O Quadro 1 faz uma adaptação do quadro oficial do *Creative Commons* e descreve todas as atribuições que cada licença possui.

| Símbolo   | Atribuição | Descrição   |
|---|------------|---|
|  | CC BY      | Esta atribuição permite a distribuição, modificação, adaptação, criação de obras derivadas, inclusive comercialização da obra.  |
|  | CC BY SA   | Esta atribuição permite a distribuição, modificação, adaptação, criação de obras derivadas, inclusive comercialização da obra, mas tudo que dela derivar deve ter o mesmo tipo de licenciamento.                  |
|  | CC BY ND   | Esta atribuição permite a distribuição, e até a comercialização da obra, no entanto ela deve estar inalterada, e na íntegra, ou seja, não é permitido a modificação, adaptação, nem a criação de obras derivadas. |

|   |                |  |
|---|----------------|--|
|  | CC BY NC       | Esta atribuição permite a distribuição, a modificação, a adaptação, e a criação de obras derivadas a partir da obra original, restringindo o uso comercial da obra original, mas as derivações não estão sujeitas as mesmas condições.                           |
|  | BY NC SA       | Esta atribuição permite modificações, adaptações, criação de obras derivadas, mas não é permitido o uso comercial da obra de origem nem de suas derivadas, sendo obrigatório que o licenciamento de derivações seja idêntico ao licenciamento da obra de origem. |
|  | CC BY NC<br>ND | Esta atribuição é a mais restritiva, pois não permite modificações, adaptações, nem uso comercial, permitindo apenas a distribuição da obra completa e download.   |

**Quadro 1 - Descrição das atribuições das licenças Creative Commons**

Fonte: Empidola, Pereira e Alves (2014, p. 6)

Essas licenças são primordiais para que um material seja REA, e dependendo do seu nível, há uma maior ou menor colaboração com o usuário.

## 2.4 AGILE

Em 2001 um grupo de profissionais muito competentes se reuniram em um hotel chamado Resort de Ski em Wasatch Range para definir quais as melhores práticas e maneiras de se desenvolver um software, desse encontro resultou num documento chamado “Manifesto Ágil” que regem o desenvolvimento ágil (BECK *et al.*, 2001), descrito no próximo tópico.

### 2.4.1 Manifesto Ágil

Segundo Gomes (2013) manifesto ágil é composto por quatro (4) valores e doze (12) princípios, sendo esses os quatro (4) valores:

1. Indivíduos e a interação entre eles mais que processos e ferramentas;
2. Software em funcionamento mais que documentação abrangente;
3. Colaboração com o cliente mais que negociação contratual;

#### 4. Responder a mudanças mais que seguir um plano.

Os doze (12) princípios do manifesto ágil, são:

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado.
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.
7. Software funcionando é a medida primária de progresso.
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade.
10. Simplicidade, a arte de maximizar a quantidade de trabalho não realizado – é essencial.
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto organizáveis.
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

#### 2.4.2 Scrum

Scrum é um dos processos mais utilizados hoje em dia, a principal característica é a *Sprint*, nome dado a cada iteração do desenvolvimento, ela pode variar de alguns dias em até um mês. Os papéis nessa metodologia são divididos em três (3) grupos, são eles: *Scrum Master*, o *Product Owner* (dono do produto) e a equipe.

*Product Owner* é o conhecedor do produto, é ele que define o que é para ser feito e prioriza as funcionalidades a serem desenvolvidas, através dessa priorização surge o *Product Backlog*, que é a lista com todas as funcionalidades a serem desenvolvidas, é ainda papel do *Product Owner* passar

todas as regras de negócio para a equipe para que a mesma transforme essas ideias em software (GOMES, 2013).

A equipe pode ser composta de 5 a 9 pessoas, das mais diversas áreas, ou seja, *cross-functional*, sendo ela formada por desenvolvedores, designers, testadores, analistas de negócios, etc, seu papel é implementar as funcionalidades definidas no *Product Backlog* até o fim estipulado da *Sprint* (SABBAGH, 2013).

O *Scrum Master* é o responsável por acompanhar o desenvolvimento da *Sprint*, cuidar para que ela aconteça, por exemplo, checar se está tudo feito de acordo e também como retirar os impedimentos da equipe, é importante destacar que o *Scrum Master* não é gerente ou chefe de equipe, porque as equipes são auto gerenciáveis/organizáveis (SABBAGH, 2013; GOMES, 2013).

A *Sprint* inicia com uma reunião estratégica para definir o que vai ser feito, quais funcionalidades serão desenvolvidas, após essa fase, parte-se para a fase de planejamento, onde é definida como será feita, ou seja, é lida tarefa por tarefa e estipulada uma estimativa para cada uma delas, somente depois dessa reunião acontece o desenvolvimento (SABBAGH, 2013; GOMES, 2013).

Ao fim da *Sprint* são feitas mais duas reuniões, a reunião de revisão, no qual é demonstrado para o *Product Owner* o que foi desenvolvido e a reunião de retrospectiva, que todos da equipe se reúnem para apontar os principais acontecimentos da *Sprint* e discutir melhorias. (SABBAGH, 2013; GOMES, 2013).

### 2.4.3 Extreme Programming

*Extreme Programming* (XP) foi criado por Kent Beck na década de 90 e é uma metodologia que cobre muito bem os aspectos técnicos do desenvolvimento, baseada em quatro (4) atividades: *ouvir, desenhar, codificar e testar*.

É preciso ouvir. Muitas vezes o programador não conhece todas as regras do negócio, por isso é feito o *Planning Game*, uma reunião para definir quais as funcionalidades a serem feitas e para que todos da equipe possam aprender sobre o que será desenvolvido e, assim, diminuir as dúvidas.

Um *design* ajuda muito o desenvolvedor a compreender o que o sistema faz e quais suas regras de negócios e as dependências, ou seja, é preciso desenhar o sistema para que todos da equipe possam entender com mais facilidade suas funcionalidades.

O XP está baseado no princípio YAGNI (*You aren't gonna need it* – Você não vai precisar disto), Gomes (2013) explica o conceito desse princípio:

Basicamente diz que devemos fazer as coisas somente no momento em que realmente precisarmos delas: muitas vezes antecipamos o desenvolvimento antes mesmo que surja a necessidade real, assumindo que um dia ela chegará, porém, essa necessidade pode nunca chegar. Ao manter o design simples e fazer somente aquilo que for necessário, você poupará tempo, porque deixará de escrever código que não precisa e terá mais tempo para se concentrar na qualidade do que real. (p. 15).

O codificar e o testar estão unidos nessa metodologia, no XP o teste de unidade é feito antes do código em produção o chamado *Test Driven Development* (TDD), ou Desenvolvimento Guiado por Testes. A programação é feita em pares e se baseia em entrega contínua, ou seja, o código é coletivo para todos os desenvolvedores.

## 2.5 HISTÓRIAS EM QUADRINHOS

Segundo Higuchi (2000) as HQs possibilitam ao leitor explorar a imaginação e discutir sobre muitos temas, como: ficção científica, magia, sonhos, violência, etc., o autor afirma ainda, que a imaginação pode aliviar os problemas sofridos no cotidiano:

[...] através da imaginação podemos superar, ou pelo menos, diminuir nossos problemas e as pressões que sofremos no cotidiano e encontrar possíveis soluções. O olhar aguçado percebe além, espírito crítico e prazer são ampliados através dele. (HIGUCHI, 2000, p. 153).

Segundo Casaril (2010) as HQs são um rico material para se trabalhar didaticamente, por serem compostas pela linguagem verbal e não-verbal (imagem e texto), sendo a interpretação feita pela relação entre as duas formas de expressão, como também o conteúdo presente nas HQs, é uma ferramenta didática que facilita o professor tratar de temas específicos e possibilita o debate em sala de aula.

Ainda, segundo a autora, as HQs possuem características que as diferenciam de outros gêneros literários, sendo elas, imagem, quadro, balão de fala, recordatório, onomatopeias, figuras cinéticas e metáforas visuais.

### 2.5.1 Pixton

É possível identificar essas características em alguns sistemas online que possuem o propósito de criar e editar HQs, como o exemplo do Pixton, único dos sistemas analisados com versão para português, criado por um casal canadense chamados Clive e Daina Goodinson, baseado



na tecnologia fechada Click-n-Drag Comics. O sistema é bastante completo, podendo editar os personagens com muita facilidade, alterando a posições de seus braços e pernas, contemplando todas as características de uma HQ definidas por Casaril (2010).

A Figura 1 apresenta a página inicial do site Pixton:



Figura 1- Página Inicial do Pixton

### 2.5.2 GoAnimate

O GoAnimate dentre os sistemas analisados é o mais completo, nesse sistema é possível criar histórias estáticas e/ou histórias animadas, possui um visual claro e intuitivo com muitas cenas e personagens distintos, é possível ainda inserir falas através de gravações feitas pelo microfone do computador, conta ainda com a possibilidade de exportar as histórias para redes sociais. O sistema contempla todas as características de uma HQ e ainda possui recursos que vão além da história estática de uma HQ convencional.

O sistema foi criado pelo Alvin Hung nos Estados Unidos da América (EUA) em 2007, conta com uma licença de software própria, porém, além dessa licença ser fechada o sistema é pago.

A Figura 2 apresenta a tela de edição/inserção de cenas.

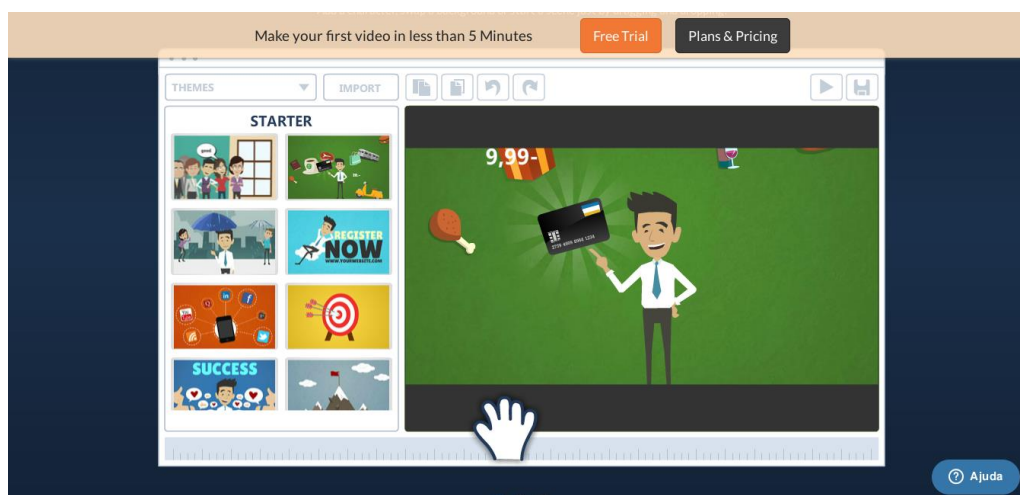


Figura 2 - GoAnimate, Tela de Inserção/Edição de Cenas

### 2.5.3 Toondoo

Foi criado pela empresa Jambav dos EUA sob uma licença fechada, porém, o sistema é grátis, dentre os analisados é o mais simples, nele é possível criar histórias estáticas e seu maior diferencial é o livro digital, onde as histórias podem ser transformadas em um livro interativo. Possui como os outros sistemas já citados todas as características de uma HQ.

A Figura 3 demonstra o livro digital de uma história chamada *Policeman*, e a página em foco é a três (3) de um grupo com cinco (5) páginas.



Figura 3 - Livro Digital do ToonDoo

### 3 MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Estará subdividido, inicialmente, em duas seções, sendo uma para os materiais e outra para o método.

#### 3.1 MATERIAIS

O Quadro 2 apresenta as ferramentas e as tecnologias utilizadas para modelar e implementar o sistema.

| Ferramenta       | Versão         | Utilidade  |
|------------------|----------------|--|
| JavaScript       | ECMAScript v.5 | Linguagem de programação de script padronizada pela ECMAScript.  |
| Node,Js          | v0.10.29       | Servidor Web baseado em JS, que utiliza o motor V8 do Google Chrome.   |
| Expressjs        | 4.12.2         | Express é um <i>framework</i> de aplicações web flexível para Node.js. Que fornece um conjunto robusto de recursos para web e aplicações móveis. |
| AngularJs        | 1.3.14         | <i>Framework</i> mantido pela Google, para o desenvolvimento de <i>single-page</i> com padrão <i>Model-View-Controller</i> .                     |
| MongoDB          | 3.0.1          | Banco de dados <i>NoSQL</i> , orientado a documentos, flexível e escalável.  |
| Atom Text Editor | v0.188.0       | Editor de texto focado para desenvolvimento web.   |

**Quadro 2 - Lista de materiais utilizados no sistema**

#### 3.2 JAVASCRIPT

A linguagem JavaScript (JS) foi criada pelo Brendan Eich na época funcionário do Netscape, sendo a mesma empresa em parceria com o Sun Microsystem criaram em 1995 a linguagem, a qual tinha o objetivo de adicionar interatividade a web, sendo a primeira versão chamada de JavaScript 1.0, em 1996 foi implementada no *browser* Netscape 2.0 (Samy, 2010).

O JS é padronizado pelo *Ecma International* pelo padrão ECMAScript, que é a especificação responsável pela padronização de *script* no lado do cliente, sendo a versão mais recente do ECMAScript é o ECMA-262 Edição 5.

Porém o JS, hoje, não executa apenas no *browser* e no lado do cliente, sendo assim, em 2009 nasce a padronização CommonJS que tem por objetivo padronizar o uso do JS fora do navegador (CommonJS, 20014).

### 3.3 MEAN

O termo MEAN foi cunhado por Valeri Karpov e emergiu em 2013 da necessidade e urgência de um amigo que precisava trabalhar com o banco de dados Postgres, foi então que Karpov teve a ideia de usar o MEAN, fazia pouco que ele e sua equipe haviam participado de um Hackthon<sup>1</sup> com duração de 24 horas e realizado uma aplicação completa com um conjunto de *frameworks*, denominado MEAN.

MEAN é a abreviação de quatro (4) tecnologias, sendo elas MongoDB (1), um banco *Not Only SQL* (NoSQL) orientado a documentos, Express (2), framework de Model-View-Controller (MVC) no lado do servidor e disponibiliza *Rest Endpoint*<sup>2</sup> para ser consumido pelo cliente, AngularJS (3), framework de MVC no lado do cliente voltado para *Single Page Web Applications*, Node.js (4), servidor baseado em JS, ou seja, MEAN é um conjunto de tecnologias para desenvolvedores *Full Stack*<sup>3</sup> em JS.

#### 3.3.1 Node.JS

Node.js surgiu no final de 2009 pelo desenvolvedor Ryan Dahl e mais 14 desenvolvedores colaboradores. A ideia para criar o sistema partiu de uma experiência do próprio Ryan ao tentar fazer *upload* de uma imagem no site Flickr. Ele ficou curioso ao saber como funcionava o “progressbar” desse *upload*, ao olhar o código percebeu que a cada quantidade de tempo definida era preciso fazer uma nova requisição para saber a porcentagem que a imagem já havia sido enviada ao servidor. Dessa experiência ele descobriu o poder das requisições *non-blocking thread* (HARRIS, 2015).

---

1 *Hackathon* significa maratona de programação. O termo resulta de uma combinação das palavras inglesas “*hack*” (programar de forma excepcional) e “*marathon*” (maratona).

2 *Endpoint* são as URL disponíveis pelo servidor que utiliza a tecnologia *REST*.

3 *Full-Stack* no mercado de trabalho é aquele que é capaz de desenvolver tanto a parte programável que é executada (*back-end*) quando você clica em um *link* quanto a construção de um *layout* de uma aplicação (*front-end*)

Segundo a definição do site do oficial, o Node.js é uma plataforma baseada no runtime JS do Chrome, para aplicações escaláveis em rede. Node.js é baseado em eventos, seu modelo é *non-blocking I/O*. O que torna ele leve e eficiente, perfeito para aplicações em tempo real e de grande volume de dados.

Segundo Junior (2014) ao descrever o funcionamento do Node.js:

O *framework* faz uso da *Javascript Engine V8* do *Google*. A *V8* é a implementação *javascript* utilizada pelo navegador *Google Chrome*, é extremamente rápida, mas necessitou de algumas modificações para que tenha um melhor funcionamento em outros contextos que não sejam o *browser*. Para que se tenha *I/O* baseado em eventos e *non-blocking*, o *Node.js* faz uso de bibliotecas *C libev* e *libeio*, desenvolvidas por Marc Lehmann.

O *Node.js* é composto por diversos módulos: módulos que fazem parte do núcleo da plataforma, chamados *core modules*, e módulos desenvolvidos pela comunidade. (p. 2)

Uma das principais características do Node.js é o *Event Loop*, a forma que o JS utiliza para lidar com os eventos de maneira mais eficaz, ele permite que o próprio *framework* gerencie as mudanças de tarefas, diferentemente de outras tecnologias nas quais quem faz esse papel é o sistema operacional, o responsável por fazer o *Event Loop* é a biblioteca *libev* (JUNIOR, 2014).

Pereira (2013) ainda discorre sobre *Event Loop*:

O *Event-Loop* é o agente responsável por escutar e emitir eventos no sistema. Na prática ele é um *loop* infinito que a cada iteração verifica em sua fila de eventos se um determinado evento foi emitido. Quando ocorre, é emitido um evento. Ele o executa e envia para fila de executados. Quando um evento está em execução, nós podemos programar qualquer lógica dentro dele e isso tudo acontece graças ao mecanismo de função *callback* do Javascript. (PEREIRA, 2013. p. 3)

### 3.3.2 Express

Express é um *framework* web *light-weight* criado em 2009 por TJ Holowaychuk (ALMEIDA, 2015), Pereira (2003) afirma que o Express surgiu para facilitar a construção de aplicações, já que a *Application Programming Interface (API) HyperText Transfer Protocol (HTTP)* nativa do Node.js seria necessário alterar muito código a cada nova alteração do sistema, sendo o Express responsável por facilitar essas mudanças constantes. O autor ainda discorre algumas características desse *framework* (p.3):

- MVR (*Model-View-Routes*);
- MVC;
- Roteamento de urls via *callbacks*;
- *Middleware*;
- Interface RESTful;

- Suporte a *File Uploads*;
- Configuração baseado em variáveis de ambiente;
- Suporte a *helpers* dinâmicos;
- Integração com *Template Engines*;
- Integração com *Structured Query Language (SQL)* e NoSQL;

### 3.3.3 AngularJS

Segundo Schmitz e Lira (2014) o AngularJS é um *framework* mantido pela Google Inc. que possuiu algumas particularidades, dentre elas, pode-se destacar que se comporta como uma extensão ao documento HTML, podendo adicionar novas *tags* e elementos ao HTML e interagindo com o JS, reduzindo assim a quantidade de código escrito.

O site oficial do AngularJS descreve em poucas palavras o que é o *framework*, é um quadro estrutural para aplicações web dinâmicas. Ele permite usar HTML como linguagem de modelo e estender a sintaxe do HTML para expressar componentes do seu aplicativo de forma clara e sucinta. A ligação de dados do AngularJS e a injeção de dependência elimina grande parte do código que deveria de ser escrito.

O AngularJS auxilia a criação de páginas *single-page*, sob o padrão MVC no lado do cliente, Schmitz e Lira (2014) afirmam que o *framework* é quase uma linguagem declarativa, por possuir tags da linguagem HTML, chamadas de diretivas, que facilitam a ligação do HTML com o JS.

### 3.3.4 MongoDB

MongoDB é um banco de dados NoSQL mantido pela empresa 10gen, escrito em C/C++, sua interface de manipulação de dados é feita em JS e a sua persistência é feita com *JavaScript Object Notation (JSON)*. É utilizado no banco o conceito de *schema-less*, ou seja, não existe relacionamento de tabelas ou chave primária, mas sim, *documents* com *embedded documents* mantidas dentro de uma *collection*. Outra característica do MongoDB é sua inserção e remoção serem executadas em runtime, evitando que uma *collection* fique travada (Pereira, 2013).

No MongoDB por não existir *joins* entre as *collection*, é possível persistir um *embedded documents* dentro de um *document*, que é o mesmo que um relacionamento entre tabelas, porém não é feita da mesma forma que nos bancos relacionais. Nesse caso é inserido o *document* da relação

dentro de outro *document*, evitando assim *selects* e *joins* desnecessários, aumentando então a velocidade na seleção dos dados (Pereira, 2013).

Nesse contexto que Almeida (2015, p. 128) discorre que “nem todas as propriedades ACID (atomicidade, consistência, isolamento e durabilidade) estão preservadas nos bancos de dados relacionais são implementadas pelo banco”, ou seja, muitas dessas propriedades precisam estar presente no desenvolvimento da aplicação e não no banco de dados.

### 3.3 MÉTODO

A construção desse material digital/ferramenta pedagógica foi dividida em quatro fases, sendo a primeira fase (1) chamada de **problemática**, nesse momento foram coletados os requisitos. Nesta etapa foi elaborado e aplicado um questionário a uma pedagoga, especialista nos estudos do fenômeno da violência sexual contra crianças e adolescentes, com o intuito de identificar possíveis problemas que esta profissional obteve ao trabalhar com vítimas de violência sexual.

A segunda fase (2) denominada de **planejamento**, foi definida juntamente com uma pedagoga. Nesta fase foram criados e analisados todos os requisitos pedagógicos e não pedagógicos para o desenvolvimento adequado do software e, a partir dos requisitos, foram elaborados os diagramas de caso de uso e de classe e o *NoSQL Data Modeling* para o banco de dados.

Na terceira fase (3), será realizado o **desenvolvimento**, momento de implementação do software. O desenvolvimento teve como base a junção de metodologias de desenvolvimento chamadas SCRUM e XP, sendo cada tarefa dividida em bloco de oito (8) horas de desenvolvimento e cada *Sprint* em uma semana, repetida em ciclos até finalizar todas as atividades.

A quarta fase (4) é o **fechamento**, marco que define a conclusão de todos os requisitos propostos na fase dois (2), feito um teste de regressão com todos os testes unitários e o lançamento da versão de produção do sistema.

## 4 RESULTADOS

Neste capítulo serão apresentados os resultados da realização deste trabalho, que é a codificação do sistema de um material didático para trabalhar com a temática de educação sexual, para prevenção da violência sexual contra crianças e adolescentes.

### 4.1 ESCOPO DO SISTEMA

O sistema tem por finalidade a construção de um material didático para se trabalhar com crianças vítimas de violência sexual e também para professores poderem trabalhar com a temática da educação sexual. O *software* será um editor de histórias em quadrinhos tornando o usuário autor de sua própria história.

O usuário poderá editar seus personagens da forma que lhe convém. O sistema dará a possibilidade de o usuário alterar as expressões faciais dos personagens, bem como suas roupas, sapatos e inserir nas cenas objetos para interagir com a história.

O sistema contará, ainda, com uma variedade de planos de fundos para que o usuário possa tornar sua história mais dinâmica e pertinente com sua imaginação. O trabalho focado na educação sexual, aparece nos personagens, que podem ser inseridos nas cenas sem roupas, personagens esses que não são assexuados, dando ao professor a possibilidade de trabalhar a educação sexual emancipatória, sem tabus e preconceitos.

### 4.2 MODELAGEM DO SISTEMA

O sistema terá como base os requisitos funcionais representados no Quadro 3. Eles representam a visão do usuário sobre o que o sistema precisa atender.

| Identificação | Nome                | Descrição   |
|---------------|---------------------|---|
| RF001         | Cadastrar histórias | O usuário do sistema poderá inserir histórias, que contém: título, descrição e licença. |
| RF002         | Cadastrar cenas     | Cadastro de cenas. Nesse cadastro será  |



|       |                      |   |
|-------|----------------------|---|
|       |                      | possível selecionar o plano de fundo.   |
| RF003 | Cadastrar usuário    | O sistema permitirá inserir usuários, sendo eles alunos/as ou professores/profissionais.                  |
| RF004 | Adicionar personagem | O usuário poderá adicionar personagens as cenas cadastradas.  |
| RF005 | Construir personagem | O usuário poderá editar seu personagem, sendo possível alterar suas expressões faciais, roupas e sapatos. |
| RF006 | Adicionar objetos    | O usuário poderá adicionar objetos as cenas cadastradas.  |

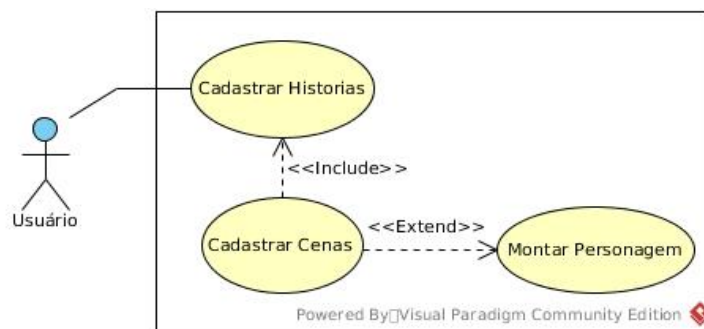
**Quadro 3 - Lista de requisitos funcionais**

A listagem do Quadro 4 representa os requisitos não funcionais do sistema, que são restrições de acesso, regras de negócio, entre outros.

| Identificação | Nome                              | Descrição   |
|---------------|-----------------------------------|---|
| RNF001        | Repositórios de imagens           | O sistema terá um repositório com as todas as imagens do sistema. |
| RNF002        | Compartilhamento em redes sociais | O usuário poderá compartilhar suas histórias em redes sociais.    |
| RNF003        | Vincular aluno ao professor       | O usuário aluno pode vincular-se ao usuário professor.            |

**Quadro 4 - Lista de requisitos não funcionais.**

A partir dos requisitos foram definidos os casos de uso apresentados na Figura 4:



**Figura 4 - Diagrama de Caso de Uso**

### 4.3 APRESENTAÇÃO DO SISTEMA

O *layout* do sistema é composto por duas telas distintas. A primeira dividida em “topo” e “conteúdo”: o “topo” apresenta o logo do sistema e a opção de login, e o “conteúdo” apresenta a composição dinâmica do sistema; o *layout* é utilizado pela tela inicial e pela tela de *login*. A segunda tela é composta por “topo”, “*sidebar*” e “conteúdo”: o “topo” apresenta o logo do sistema e as informações de usuários, o “*sidebar*” possui várias opções de acesso e as funcionalidades do sistema, e o “conteúdo” apresenta as informações das opções selecionadas no “*sidebar*”; o *layout* é utilizado por todas as outras telas do sistema.

A tela de “login” possibilita duas formas de acesso ao sistema. Uma delas é por meio das redes sociais, há três (3) formas de acesso via redes sociais: (1) Github; (2) Facebook; (3) Google. A outra é por meio de um cadastro “local”, no qual obtém-se a informação de: *login*, nome, sobrenome, e-mail e senha.

A Figura 5 possui a opção da tela “Criar Quadrinho”, há também, na parte esquerda da tela, as opções de imagens a serem inseridas na história do quadrinho, e o menu de seleção de imagens que contém as seguintes opções: fundos, personagens e objetos. A opção “imagens de fundo” diz respeito às imagens que farão parte do *background* de toda história, a “imagens de personagens” faz referência aos personagens disponíveis pelo sistema para inserir na história, divididas em masculino e feminino, e a opção “objetos” apresenta todos os objetos que não fazem parte de um personagem, mas que podem ser inseridos na tela juntamente com os balões de fala.

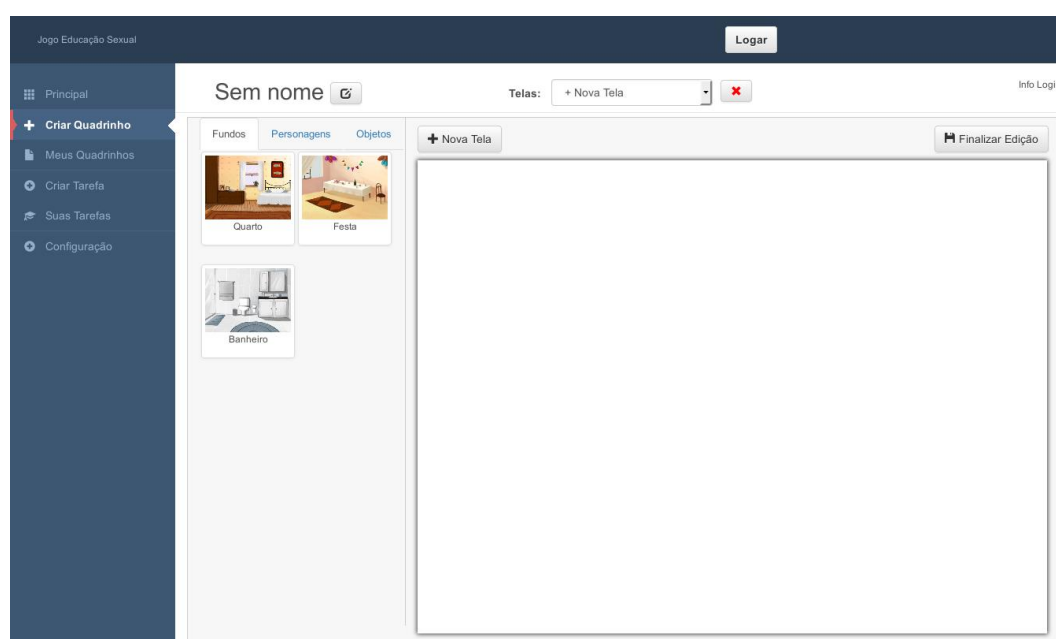
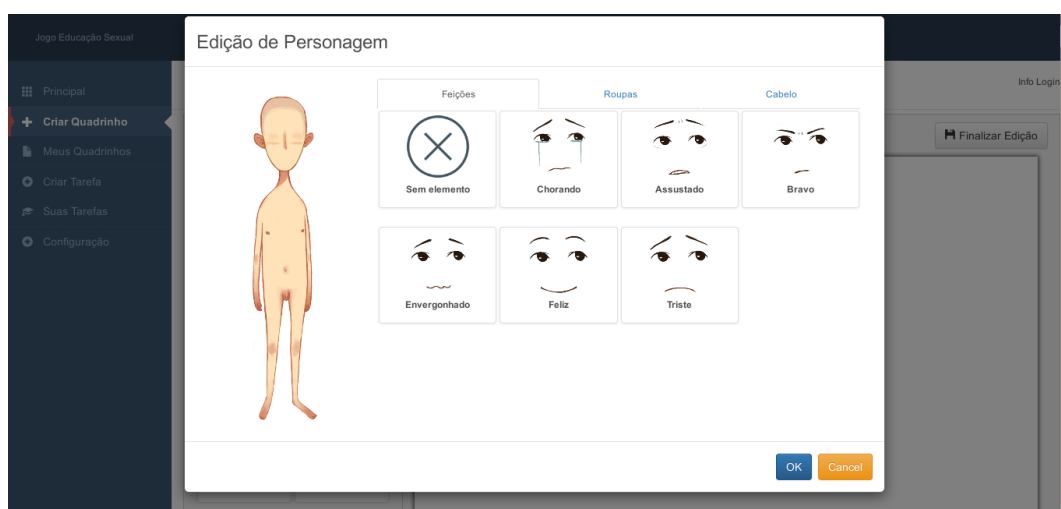


Figura 5 - Tela Criar Quadrinho

Ainda na Figura 5, no topo da página de conteúdo há a opção “nome da cena”, como padrão o nome apresentado é “Sem Nome”, ao lado há um botão para alterar o nome, no centro do topo do conteúdo contém um *select box* para alterar entre as cenas criadas ou adicionar uma nova cena e logo em seguida há o botão para remover a cena selecionada no *select box*.

Ao lado do menu de imagens e abaixo do topo do conteúdo, existe a tela para edição da cena, Na parte acima dessa área existem dois botões, sendo o primeiro com o nome de “Nova Tela” o qual salva a tela recém criada e insere uma nova tela para edição. O segundo com o nome “Finalizar Edição” que tem a função de inserir as cenas já criadas no banco de dados.

Para que se insira um fundo, um personagem ou um objeto à cena, é preciso arrastar a imagem no “menu de imagens” até a “área de edição”. Caso a imagem arrastada é um fundo ou um objeto (exceto os balões de fala) a imagem será inserida após ser solta do evento *drag-and-drop*, caso for um personagem, ao invés de ser inserida na cena é aberta a tela de “Edição de Personagem”, tela essa representada pela Figura 6.

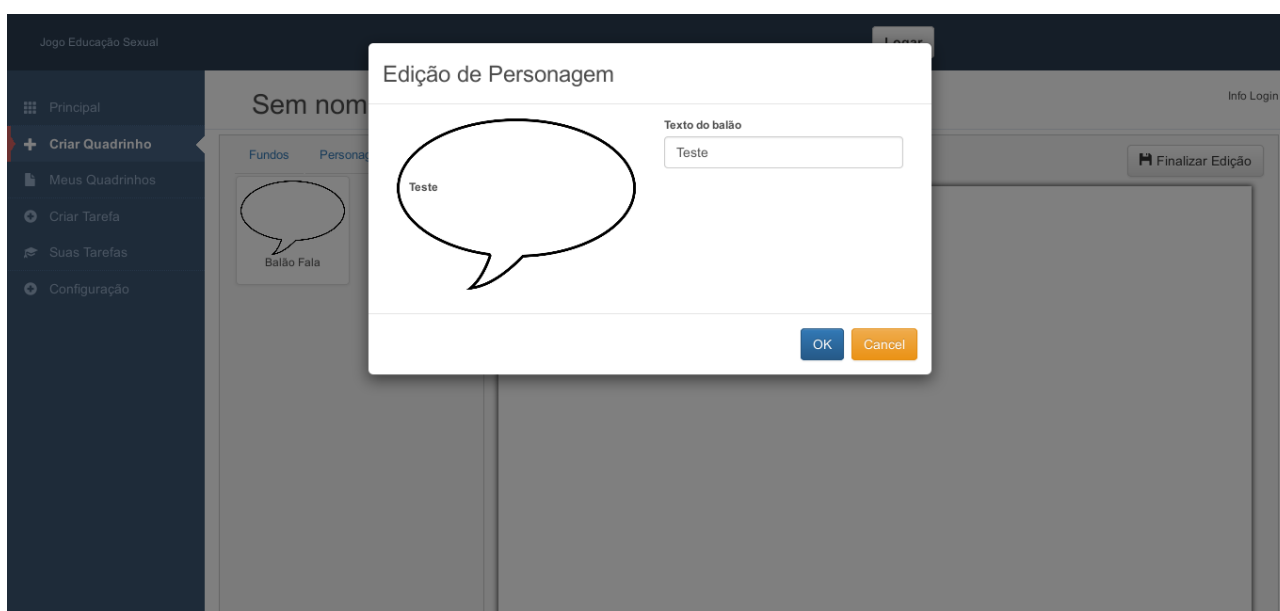


**Figura 6 - Tela Edição de Personagem**

No lado esquerdo da tela “Edição de Personagem” está a imagem de *preview* do personagem, no lado direito, o menu de opções e seus respectivos conteúdos. Cada personagem contém três (3) opções que podem ser selecionadas: “feições”, “roupas” e “cabelo”. As “feições” são todas as expressões faciais que se pode adicionar ao personagem, sendo elas: chorando, assustado, bravo, feliz, dentre outras; as “roupas” são todas as vestimentas disponíveis para inserir no personagem, como: sapatos, tênis, entre outros; o “cabelo” são todos os cabelos disponíveis para se inserir ao personagem, e estão divididos em: liso, enrolado, curto e comprido, podendo escolher as cores.

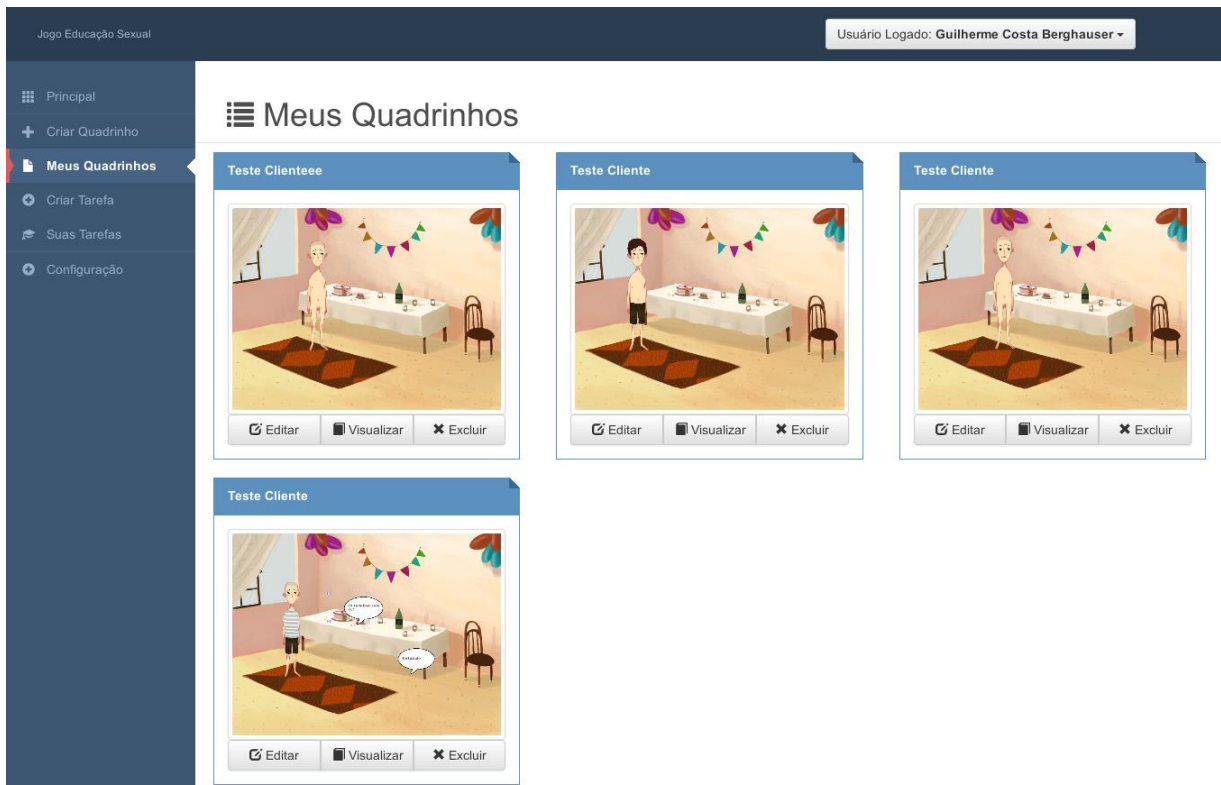
Para que se insira ao personagem algum dos elementos citados, basta um clique sobre o mesmo que automaticamente a imagem de *preview* será atualizada com o objeto selecionado. Após a edição, na parte inferior da tela há dois botões, o “OK” que inserirá na cena o personagem editado e simultaneamente fechará a tela de edição, e o “Cancelar” que fechará a tela e não realizará nenhuma edição ou alteração.

Ao inserir um balão de fala, a regra de negócio é muito parecida com a edição de personagem, porém a tela de edição apenas contém a imagem de *preview* e um *input* para adicionar o texto que estará contido nesse balão, regra representada pela Figura 7. Para alterar um elemento já inserido à cena é preciso dar um duplo clique no elemento editável e a sua respectiva tela de edição se abrirá.



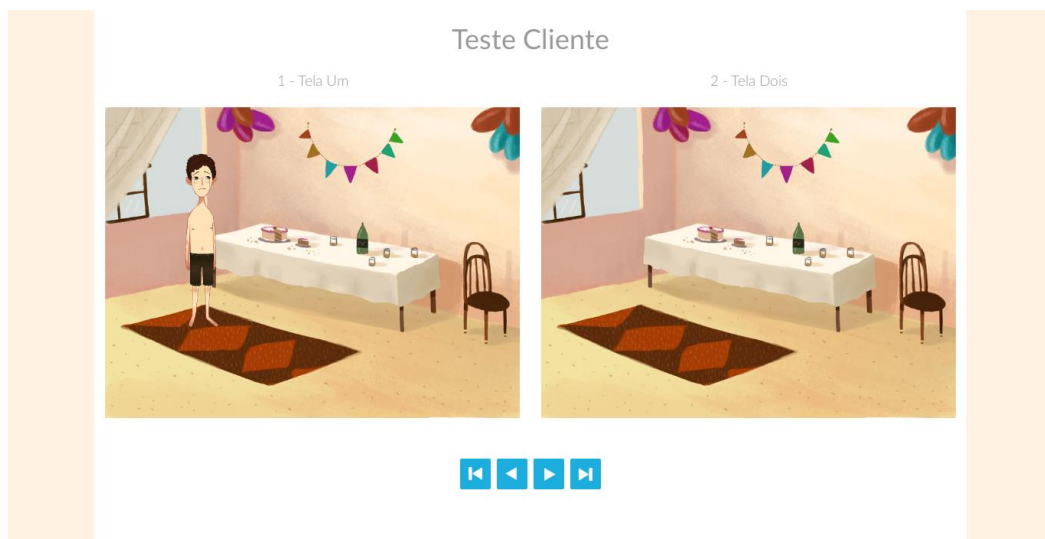
**Figura 7 - Edição de Balão de Fala**

Na tela “Meus Quadrinhos”, representada pela Figura 8 é exibida uma lista com todas as histórias já criadas e em cada uma delas é apresentada a imagem da primeira cena dessa história e, logo abaixo, estão dispostos três (3) botões, o primeiro (1) é o “Editar”, ao clicar sobre ele a tela “Criar Quadrinho” é aberta com todas as cenas da história.



**Figura 8 - Meus Quadrinhos**

O segundo (2) é o “Visualizar” ao clicar sobre ele é aberta uma nova janela ou aba do navegador com as cenas em modo de leitura, esse modo é apresentado na Figura 9, e o terceiro (3) é o botão “Excluir” que tem por finalidade excluir a história.



**Figura 9 - Tela Modo de Leitura**

Ainda na tela “Meus Quadrinhos” ao clicar sobre a imagem de exibição de uma história, é aberta a tela para edição de informações da história, nela contém o campo para alterar o nome da história e/ou adicionar essa história a uma tarefa através de uma *hash*.

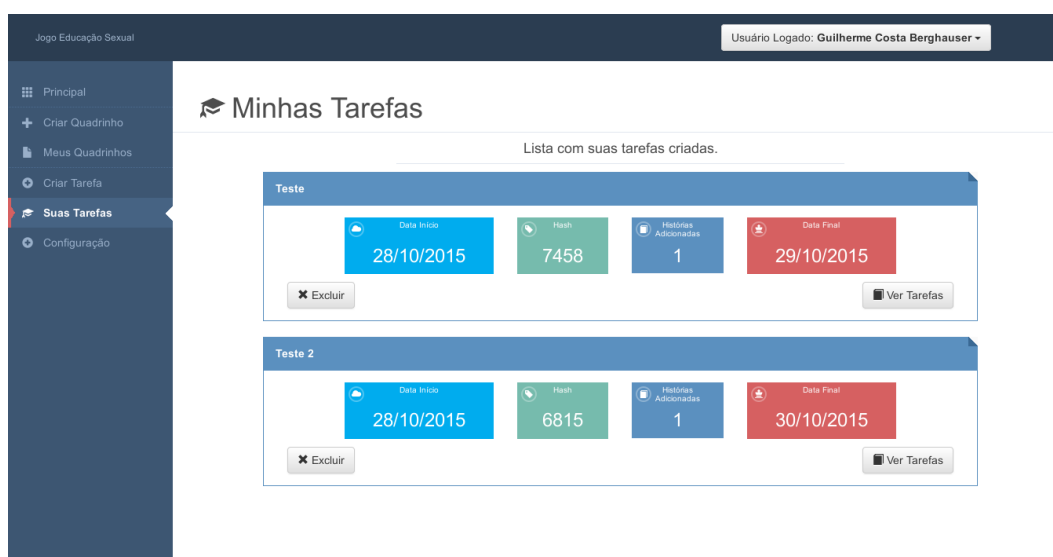
A tela “Criar Tarefa”, exibida na Figura 10, tem por finalidade gerar uma *hash* para que outros usuários possam adicionar suas histórias a tarefa criada, nela são apresentados dois campos, o primeiro é o nome da tarefa e o segundo é a data-limite para o cadastro das histórias à tarefa, ao inserir essas informações e salvar, o sistema gerará a *hash* para que possa compartilhar com outros usuários.



The screenshot shows the 'Criar Tarefa' interface. At the top, it says 'Jogo Educação Sexual' and 'Usuário Logado: Guilherme Costa Berghauer'. The left sidebar has options: Principal, Criar Quadrinho, Meus Quadrinhos, Criar Tarefa (selected), Suas Tarefas, and Configuração. The main area is titled 'Criar Tarefa' and contains the instruction: 'Para gerar a hash de compartilhamento basta salvar a tarefa.' Below this are three input fields: 'Nome da Tarefa' (empty), 'Data Final Tarefa' (containing '27/10/2015'), and 'Hash de compartilhamento' (containing 'Hash'). A 'Salvar' button is positioned below the 'Data Final Tarefa' field.

Figura 10 - Tela Inserir Tarefa

Na Figura 11 é exibida a tela “Minhas Tarefas”, nela é apresentada uma lista com todas as tarefas criadas pelo usuário, exibindo as informações contidas nestas tarefas, como: data de criação da tarefa, a *hash* da tarefa, quantidade de histórias já adicionadas a tarefa e data-limite para enviar histórias da tarefa. Possui ainda dois botões, um para mostrar os links de todas as histórias adicionadas e outro para excluir a tarefa.



The screenshot shows the 'Minhas Tarefas' interface. At the top, it says 'Jogo Educação Sexual' and 'Usuário Logado: Guilherme Costa Berghauer'. The left sidebar has options: Principal, Criar Quadrinho, Meus Quadrinhos, Criar Tarefa, Suas Tarefas (selected), and Configuração. The main area is titled 'Minhas Tarefas' and contains the instruction: 'Lista com suas tarefas criadas.' Below this are two task cards. The first card is titled 'Teste' and shows: 'Data Início: 28/10/2015', 'Hash: 7458', 'Histórias Adicionadas: 1', and 'Data Final: 29/10/2015'. The second card is titled 'Teste 2' and shows: 'Data Início: 28/10/2015', 'Hash: 6815', 'Histórias Adicionadas: 1', and 'Data Final: 30/10/2015'. Each card has an 'Excluir' button on the left and a 'Ver Tarefas' button on the right.

Figura 11 - Tela Minhas Tarefas

A tela de “Configuração” contém as informações do usuário, como: nome, e-mail e tipo de usuário, tais informações podem ser alteradas, caso o usuário efetue o login através das redes sociais pode acontecer do nome ou e-mail não serem preenchidos automaticamente, fazendo-se necessário exibir a tela “Configuração” após o login para finalizar o cadastro de usuário. A tela de configuração é exibida na Figura 12.

The screenshot shows the 'Edição de Usuário' (User Edit) page. At the top, it says 'Jogo Educação Sexual' and 'Usuário Logado: Guilherme Costa Berghauer'. The sidebar on the left has a menu with 'Configuração' highlighted. The main content area has a title 'Edição de Usuário' and a message: 'Para continuar a utilizar o aplicativo, termine seu cadastro.' Below this, there are two input fields: 'Nome' with the value 'Guilherme Costa Berghauer' and 'Email' with the value 'tortoyoyo@gmail.com'. To the right, there is a 'Tipo de Usuário' section with radio buttons for 'Alunx', 'Professor/a' (which is selected), 'Psicólogo', 'Conselheirx Tutelar', and 'Assistente Social'. At the bottom center, there is a 'Salvar' button.

**Figura 12 - Configuração**

Importante ressaltar que todo o software está licenciado sob a GPL 3.0, ou seja, é um software livre, tornando acessível seu código a qualquer desenvolvedor ou ainda seu uso em qualquer situação, bem como as histórias desenvolvidas pelos usuários estão licenciados pela licença CC BY, a licença mais permissivas disponíveis pela *Creative Commons*.

#### 4.4 IMPLEMENTAÇÃO DO SISTEMA

Na programação com MEAN é importante diferenciar o cliente do servidor, ambos são estruturados no modelo MVC, na Listagem 1 é possível ver como é setado a rota de URL no servidor e como o Express constrói um *Web Service REST*.

```

1 module.exports = function(app) {
2   var util = app.controllers.util;
3   var controller = app.controllers.usuario;
4   app.route('/usuario').get(util.verificaAutenticacao, controller.getUsuario).put(util.verificaAutenticacao,

```

```

    controller.setUsuario);
5   };

```

### Listagem 1 – Rota de URL do Usuário

Na linha um (1) o sistema faz a injeção do código, o módulo utilizado para essa injeção é chamado de Require.js, empregado desde a primeira versão do Node.js. A variável “app” permite que se acesse qualquer código injetado anteriormente ao servidor. Na linha dois (2) a variável “util” passa a ter todos os métodos do arquivo util.js da pasta /app/controllers/, com a variável “controller” se repete a injeção só que agora assumindo todos os métodos do arquivo usuario.js da pasta /app/controllers/.

Na linha seguinte, o método “app.route()” define a rota para requisição no servidor, no exemplo a rota “/usuario” foi setada, ou seja, o endereço “http://endereco:porta/usuario” estará disponível. Os métodos seguintes “get()” e “put()” definem os *controllers* aos cabeçalhos HTTP para requisição. No método “get()” inicialmente há a verificação para constatar se o usuário está logado ao sistema, caso a resposta seja positiva executará o método “getUsuario()”, o mesmo acontece com o método “put()”, nesse caso o método chamado é o “setUsuario()”.

```

1     var sanitize = require('mongo-sanitize'); // EVITAR SQL INJECT
2
3   module.exports = function(app) {
4     var Usuario = app.models.usuario;
5     var controller = {};
6
7     controller.getUsuario = function(req, res) {
8       res.json(req.user);
9     };
10    controller.setUsuario = function(req, res) {
11      var novo = sanitize(req.body);
12      var id = novo._id;
13      console.log(req.body);
14      delete novo._id;
15      Usuario.findByIdAndUpdate(id, novo, {
16        new: true,
17        upsert: true
18      },
19      function(erro, usuario) {
20        if (!erro) {
21          console.log(usuario);
22          res.json(usuario);
23        } else {
24          console.error(erro);
25          res.status(500).json(erro);
26        }
27      }
28    );
29  };
30 };
31

```



```

32   return controller;
33 };

```

### Listagem 2 – Controller do Usuário

A Listagem 2 exemplifica o *controller* utilizado pelas rotas descritas pela Listagem 1. Na linha um (1) tem-se a injeção do “mongo-sanitize”, módulo utilizado pelo Node.js para evitar algo parecido com o *SQL Injection* nos bancos NoSQL, na linha seguinte ao carregar o arquivo `module.exports` as funções do mongoose dentro da *collections* “Usuário” é carregada, o mongoose é o drive utilizado pelo Express para conectar o Node.js ao MongoDB.

O método “`controller.getUsuario`” retorna ao cliente o JSON contendo informações do usuário serializado na seção da conexão, o método “`controller.setUsuario`” faz a alteração do usuário no banco de dados, na primeira linha ele recupera da requisição as informações enviadas pelo cliente por meio da variável “`req.body`”, o método “`Usuario.findByIdAndUpdate`” faz a busca do objeto pelo campo *id* e em seguida o altera com as informações contidas na variável “`novo`”, na função de retorno, caso tenha um erro o servidor responde ao cliente o status 500 e o JSON com as informações do erro, caso contrário envia um JSON com as informações do usuário alteradas.

A Listagem 3 exemplifica a tela no lado do cliente para o consumo do REST descritos nas Listagens 1 e 2.

```

1      <div class="row titulo">
2      <div class="col-md-12">
3      <h1><span class="glyphicon glyphicon-user" style="font-size:      30px;"></span> Edição de
Usuário</h1>
4      </div>
5      </div>
6
7      <div class="row">
8      <div class="col-md-6 col-md-offset-3 text-center box-border-bottom">
9      <h4> Para continuar a utilizar o aplicativo, termine seu cadastro.</h4>
10     </div>
11     </div>
12
13     <div class="row" style="margin-top: 10px !important;">
14     <div class="col-md-5 col-md-offset-1 box-border-left">
15     <div class="form-group">
16     <label for="exampleInputEmail1">Nome:</label>
17     <div class="input-group">
18     <span class="input-group-addon">Nome:</span>
19     <input class="form-control" ng-model="user.nome" type="text">
20     </div>
21     </div>
22
23     <div class="form-group">
24     <label for="exampleInputEmail1">Email</label>
25     <div class="input-group">
26     <span class="input-group-addon">@</span>

```

```

27     <input class="form-control" ng-model="user.email" type="text">
28   </div>
29 </div>
30
31 </div>
32
33 <div class="col-md-5 col-md-offset-1 box-border-left">
34   <div class="form-group">
35     <label for="exampleInputEmail1">Tipo de Usuário</label>
36   </div>
37   <div class="radio">
38     <label>
39       <input type="radio" name="radioProfissao" value="aluno" ng- model="user.tipoUsuario">
40       Alunx
41     </label>
42   </div>
43   <div class="radio">
44     <label>
45       <input type="radio" name="radioProfissao" value="professor" ng- model="user.tipoUsuario">
46       Professor/a
47     </label>
48   </div>
49   <div class="radio">
50     <label>
51       <input type="radio" name="radioProfissao" value="psicologo" ng-model="user.tipoUsuario">
52       Psicólogx
53     </label>
54   </div>
55   <div class="radio">
56     <label>
57       <input type="radio" name="radioProfissao" value="conselho" ng- model="user.tipoUsuario">
58       Conselheirx Tutelar
59     </label>
60   </div>
61   <div class="radio">
62     <label>
63       <input type="radio" name="radioProfissao" value="assistentesocial" ng-
64       model="user.tipoUsuario">
65       Assistente Social
66     </label>
67   </div>
68 </div>
69
70 <div class="row" style="margin-top: 10px !important;">
71   <div class="col-md-6 col-md-offset-3 box-border-top">
72     <button type="button" style="margin-top:5px;" ng-click="updateUsuario()" class="btn btn-default center-
73     block">Salvar</button>
74   </div>
75 </div>

```

**Listagem 3 – Tela cadastro usuário**

O exibido na Listagem 3 é uma estrutura de HTML padrão, o que diferencia para uso no AngularJS são as *tags* adicionais “ng-”, no exemplo é utilizado o “ng-model” que diz ao AngularJS qual é a variável no *controller* da tela que o elemento vai ser preenchido.

Na Listagem 4 é apresentado o *controller* da tela demonstrada na Listagem 3:

```

1  angular.module('jogo').controller('CadUsuarioController',
2      function($scope, $resource, $routeParams, $location, initPage, Usuario) {
3
4
5      $scope.init = function() {
6
7          var init = {
8              menu: 'configuracao'
9          };
10
11         initPage.pageCompleta($scope, init);
12
13         Usuario.getUsuarioLogado(function(res) {
14             $scope.user = res;
15             if($routeParams.tela == 1 && $scope.user.email &&
16                 $scope.user.tipoUsuario != " && $scope.user.nome){
17                 $location.path('/principal');
18             }
19         });
20     };
21     $scope.init();
22
23     $scope.updateUsuario = function(){
24         Usuario.updateUsuario($scope.user, function(res){
25             $scope.user = res;
26             swal("Usuário Alterado", "Usuário alterado com sucesso!", "success");
27         });
28     };
29
30 });

```

### Listagem 3 – Controller usuário

Na linha um (1) é carregado o *module* da aplicação (“jogo”), em seguida é sobrecarregado o método “controller()”, seu primeiro parâmetro é o nome do *controller* e, o segundo parâmetro é a função com a injeção das dependências.

Toda variável ou função inserida na dependência “\$scope” ficará visível para *view*, caso esse valor seja alterado no *controller* o AngularJs faz a alteração na *view* automaticamente, agilizando assim o desenvolvimento.

O método “\$scope.init()” executa a função para alterar as classes no elemento menu da *view*, essa função adiciona a classe “actived” na opção “Configuração” do menu, alterando a aparência do elemento e, logo em seguida busca no banco as informações do usuário. No retorno dessa função é verificado se todos os campos estão preenchidos, caso a resposta seja positiva ele sistema redireciona o usuário para página principal. A dependência “\$location” é responsável por fazer a troca de página.

Por último, a função “\$scope.updateUsuario()” é encarregada de fazer a alteração das informações do cliente. Na Listagem 3 na linha 72 é possível ver a tag “ng-

click="updateUsuario()" inserida no elemento *button*, e essa *tag* informa ao AngularJs qual função será executada quando o evento de clique for disparado. A dependência “Usuario” é criada por meio de um “*factory*” que tem suas características muito parecidas com uma classe estática do Java, ao ser carregada para o Angular ela pode ser acessada em qualquer *controller* ou *services* do Angular sem que aloque um novo espaço em memória.

Na Listagem 4 é demonstrado o *factory* “Usuario” e suas funções:

```

1  angular.module('jogo').factory('Usuario', ['$resource', 2    function($resource) {
3      var usuario = $resource('/usuario', {}, {
4        'query': {
5          method: 'GET',
6          isArray: false
7        },
8        'update': {
9          method: 'PUT'
10       }
11     });
12
13     var retorno = {
14
15       getUsuarioLogado: function(callback) {
16         if (!usuario._id) {
17           usuario.query(function(elem) {
18             setUsuario(elem);
19             callback(usuario);
20           }, function(err) {
21             callback(err);
22           });
23
24         } else {
25           callback(usuario);
26         }
27       },
28       updateUsuario: function(usuarioNovo, callback) {
29         usuarioNovo.$update(function(elem) {
30           setUsuario(elem);
31           callback(elem);
32         });
33       }
34     };
35
36     var setUsuario = function(usuarioLocal){
37       usuario = usuarioLocal;
38     };
39
40     return retorno;
41   });

```

Listagem 4 – Factory usuário

O primeiro parâmetro no método “factory()” define o nome que o serviço terá, o segundo parâmetro é a função que injeta suas dependências, demonstrado nesse exemplo como “\$resource”,

essa dependência tem a finalidade de auxiliar o desenvolvedor, proporcionando maior facilidade na execução de requisições à um *Web Service* REST.

Na primeira linha após a configuração do *factory* é possível observar uma variável chamada “usuario”, nela é instanciado o “\$resource”, sendo o primeiro parâmetro dessa instância a URL do *Web Service*. No caso do exemplo “/usuario”, no parâmetro seguinte é explicitado que a função “query” utilizará o cabeçalho “GET” e seu retorno será um *Array*, já o método “update” utilizará o cabeçalho “PUT”. Para que os métodos e variáveis criados dentro do *factory* sejam visíveis aos *controllers* é preciso que esse método esteja no JSON do retorno da função, caso não esteja ele será um método ou variável privada a esse *factory*.

O método “getUsuarioLogado()” executará o método “query()” que será responsável pela requisição “GET” na URL “/usuario” e, assim que o servidor retornar as informações o método de *callback* será chamado. O método “updateUsuario()” fará uma requisição “PUT” para a URL “/usuario” e todos os dados contidos na variável “usuario” serão enviadas ao *Web Service*, o retorno será o objeto com as novas informações de usuário e executará o método de *callback*.

## 5 CONCLUSÃO

A WEB 2.0 trouxe muitos benefícios não somente para os usuários, mas para todos os ramos da sociedade. Nesse trabalho foi destacado o surgimento da educação aberta por meio dessa mudança de perfil dos usuários da internet, e ainda mais, foi discutido sobre um novo conceito de educação no âmbito da Internet, com emprego de REA, e seus benefícios para a formação de pessoas críticas.

Ainda foi discorrido o conceito de violência, e dado ênfase na violência sexual infantil, fenômeno esse muito triste e atormentador em nossa sociedade. A partir desse contexto foi proposto o desenvolvimento de um material REA para professores ou profissionais que lidam com crianças vítimas de violência sexual, que tem por objetivo ajudar essas crianças a discutirem sua sexualidade de forma emancipatória, por meio da educação emancipatória.

No âmbito tecnológico, foi discorrido sobre as tecnologias focadas em JavaScript, como Node.js, Express e AngularJS, assim também, o banco de dados MongoDB, tecnologias recentes e muito promissoras, todas elas criadas com foco na escalabilidade do aplicativo e de fácil manuseio.

Como resultado deste trabalho foi desenvolvido um software de edição de histórias em quadrinhos para profissionais que trabalham com crianças e adolescentes vítimas de violência sexual, para que possam identificar a violência cometida sobre essas crianças ou adolescentes, bem como trabalhar com a temática sexualidade em sala de aula ou em alguma atividade de educação sexual visando a prevenção da violência sexual.

Na elaboração do trabalho a utilização da tecnologia AngularJs foi em alguns momentos algo assertivo e em outros foi algo muito complexo, ele tem por característica facilitar muito a alteração dos elementos no DOM, porém como o trabalho possui muitos elementos para se alterar na *view* é preciso criar muitas *directives* tornando algo muito trabalhoso e aumentando muito a quantidade de linhas de código do sistema.

Com esses problemas os próximos passos serão: (1) trocar o método de trabalhar na *view* do AngularJs para React, tecnologia essa criada pelo Facebook, tornando assim o trabalho com as imagens mais performático, (2) reestruturar um novo layout junto a um profissional de design, pensando na usabilidade do usuário, tornando assim o sistema fácil de usar, (3) utilizar canvas ao invés de imagens .png, podendo assim usufruir da tecnologia WebGL, melhorando a renderização das imagens.

## REFERÊNCIAS

ADORNO, Theodoro W. **Educação e emancipação**. Rio Janeiro: Paz e Terra, 2006.

ALMEIDA, Flávio. **Mean: Full stack JavaScript para aplicações web com MongoDB, Express, Angular e Node** – 1 ed. São Paulo: Casa do Código, 2015.

ARENDT, Hannah. **Sobre a violência**. Trad. André Duarte. Rio de Janeiro: Relume-Dumará, 1994.

AZEVEDO, Maria Amélia; GUERRA, Viviane Nogueira de Azevedo (Org.). **Crianças vitimizadas: a síndrome do pequeno poder**. – 2ª ed., São Paulo: Iglu, 2007.

BECK, Kent; BEEDLE, Mike; BENNEKUM, Arie van; COCKBURN, Alistair; CUNNINGHAM, Ward; FOWLER, Martin; GRENNING, James; HIGHSMITH, Jim; HUNT, Andrew; JEFFRIES, Ron; KERN, Jon; MARICK, Brian; MARTIN, Robert C.; MELLOR, Steve; SCHWABER, Ken; SUTHERLAND, Jeff; THOMAS, Dave. **Manifesto for agile software development**. Disponível em: <<http://agilemanifesto.org>>. Acessado em: 10 de julho de 2015.

BRASIL. **Lei nº. 9.610, de 19 de fevereiro de 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências**. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, 20 fev. 1998. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/l9610.htm](http://www.planalto.gov.br/ccivil_03/leis/l9610.htm)>. Acesso em: 10 de junho de 2015.

CASARIL, Marina. **Histórias em Quadrinhos: capacidades de leitura na relação entre o verbal e o não-verbal observadas em simulados da prova brasil**. Trabalho de Conclusão de Curso apresentado ao Colegiado do Curso de Letras – Português/Inglês e respectivas Literaturas para obtenção do título de Licenciado em Letras. Cascavel-PR, 2010.

CHAUÍ, Marilena. **Repressão sexual: essa nossa (des)conhecida**. 6. ed. São Paulo: Brasiliense, 1984.

CUNNINGHAM, A. H.; SAS, L. D. **The role of school programs in disclosure. Tipping the balance to tell the secret: The public discovery of child sexual abuse**. Ontario: London Family Court Clinic Inc, 1995.

DECKER, Isabel Cristina Uarthe. **A Categoria Emancipação em Paulo Freire e suas Contribuições para um Processo de Educação Sexual Emancipatória**. – Florianópolis, 2010. 200 p.

EMPIDOLA, Joice de; PEREIRA, Angela Maria de Almeida; ALVES, Thelma Panerai. **Competência Digital E Possibilidades De Colaboração Com Recursos Educacionais Abertos (REAs) - Encontro de Pesquisa Educacional em Pernambuco, 2014**.

FALEIROS, Vicente de Paula; FALEIROS, Eva T. Silveira. **Escola que protege: enfrentando a violência contra crianças e adolescentes**. 2 ed. Brasília: Ministério da Educação, Secretaria de Educação Continuada, Alfabetização e Diversidade, 2008. Disponível em: <[http://portal.mec.gov.br/secad/arquivos/pdf/escqprote\\_eletronico.pdf](http://portal.mec.gov.br/secad/arquivos/pdf/escqprote_eletronico.pdf)>. Acesso em: 10 de Junho de 2015.

FIGUEIRÓ, Mary Neide Damico. **Formação de Educadores Sexuais: adiar não é mais possível**. – Campinas, SP: Mercado de Letras; Londrina, PR: Eduel. (Coleção Dimensões da Sexualidade), 2006.

GAGLIOTTO, Giseli Monteiro. **A Educação Sexual na Escola e a Pedagogia da Infância: matrizes institucionais, disposições culturais, potencialidades e perspectivas emancipatórias**. Tese (Doutorado em Pedagogia). – Campinas, SP, 2009.

GAGLIOTTO, Giseli Monteiro; LEMBECK, Tatiane. **Sexualidade e Adolescência: a educação sexual numa perspectiva emancipatória**. Educere et Educare – Revista de Educação ISSN: 1981-4712 (eletrônica) – 1809-5208 (impressa) Vol. 6 – No 11 – 1º Semestre de 2011.

GOMES, André Faria. **Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio**. 1 ed. São Paulo: Casa do código, 2013.

HARRIS, Amber. **The Birth of Node: Where Did it Come From? Creator Ryan Dahl Shares the History**. Disponível em: <<http://siliconangle.com/blog/2013/04/01/the-birth-of-node-where-did-it-come-from-creator-ryan-dahl-shares-the-history/>> Acesso em: 10 de junho de 2015.

HIGUCHI, Kazuko Kojima. **Aprender e ensinar com textos não escolares**. 3. ed. São Paulo: Cortez Super. Homem, Mônica & Cia. CHIAPPINI, Lígia. (org.). 2000.

JUNIOR, Francisco de Assis Ribeiro. **Programação Orientada a Eventos no lado do servidor utilizando Node.js**. Disponível em: <[http://www.infobrasil.inf.br/userfiles/16-S3-3-97136-Programa%C3%A7%C3%A3o%20Orientada\\_\\_\\_\\_.pdf](http://www.infobrasil.inf.br/userfiles/16-S3-3-97136-Programa%C3%A7%C3%A3o%20Orientada____.pdf)>. Acessado em: 10 de Junho de 2015.

KARPOV, Valeri. **The MEAN Stack: MongoDB, ExpressJS, AngularJS and Node.js**. Disponível em: <<http://blog.mongodb.org/post/49262866911/the-mean-stack-mongodb-expressjs-angularjs-and>>. Acesso em: 10 de Junho de 2015.

LANDINI, Tatiana Savoia. **O professor diante da violência sexual**. – São Paulo: Cortez. – (Coleção educação e saúde; v.4), 2011.

MORAIS, Carlos; MIRANDA, Luísa; DIAS, Paulo - **Web 2.0 e recursos educacionais abertos. Tecnologia Educacional**. ABT – Tecnologia em Educação. 2013. Disponível em: <[https://bibliotecadigital.ipb.pt/bitstream/10198/10091/3/RV\\_2013\\_Web20\\_RecEducAberAprendizagem.pdf](https://bibliotecadigital.ipb.pt/bitstream/10198/10091/3/RV_2013_Web20_RecEducAberAprendizagem.pdf)>. Acesso em: 10 de junho de 2015.

NUNES, César A. **Filosofia, sexualidade e educação: as relações entre pressupostos ético-sociais e históricos-culturais presentes nas abordagens institucionais sobre a educação sexual escolar**. 1996. Tese (Doutorado em Educação)- Faculdade de Educação, Universidade Estadual de Campinas, 1996.



PEREIRA, Caio Ribeiro – **Aplicações Web realtime com Node.js**. 1 ed. São Paulo: Casa do Código, 2013.

PERKINS, D.D.; ZIMMERMAN, M.A. **Empowerment meets narrative: listening to stories and creating settings**. AMERICAN JOURNAL OF COMMUNITY PSICOLOGY. v. 23. n. 5. Disponível em: <<http://dpss.psy.unipd.it/link/files/3-Rapport.1995.Empowerment.meets.narrative.AJCP1.pdf>>. Acesso em: 10 de junho de 2015.

POGREBINSCH, Thamy. **Emancipação: um conceito em busca de uma teoria**. In: ENCONTRO NACIONAL DA ABCP - ASSOCIAÇÃO BRASILEIRA DE CIÊNCIA POLÍTICA, 4. Rio de Janeiro. *Anais eletrônico...* Rio de Janeiro, 2004. Área Teoria Política. Painel 2. Disponível em: <[http://www.cienciapolitica.org.br/abcp/arquivos/programa\\_new.pdf](http://www.cienciapolitica.org.br/abcp/arquivos/programa_new.pdf)>. Acesso em: 10 de Junho de 2015.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. 1 ed. São Paulo: Casa do código, 2013.

SANTAROSA, Lucila Maria Costi; CONFORTO, Debora; SCHNEIDER, Fernanda Chagas - **Tecnologias na Web 2.0 [Em linha]: o empoderamento na educação aberta**. In Colóquio Luso-Brasileiro de Educação a Distância e Elearning, 3, Lisboa, 2013. "Colóquio Luso-Brasileiro...: atas". Lisboa: Universidade Aberta. LEAD, 2014.

SANTOS, A. I. (2012). **Educação aberta: histórico, práticas e o contexto dos recursos educacionais abertos**. In N. Pretto, C. Rossini, & B. Santana (Eds.), Recursos Educacionais Abertos: práticas colaborativas e políticas públicas. Casa da Cultura Digital e Comitê Gestor da Internet no Brasil.

\_\_\_\_\_. Rita Ippolito – Seropédica, RJ: EDUR, 2011. Disponível em: <<http://portaldoprofessor.mec.gov.br/storage/materiais/0000016936.pdf>>. Acesso em: 10 de Junho de 2015.

SCHMITZ, Daniel; LIRA, Douglas. **AngularJS na prática - Crie aplicações web com AngularJS**. 1 ed. São Paulo, 2014.

SILVA, Maurício Samy. **JavaScript : guia do programador**. 4 ed. São Paulo: Novatec Editora, 2010.

SILVA, Plácido. **Vocabulário jurídico**. Rio Janeiro: Companhia Editora Florense, 2005.

UNESCO. **Declaração Rea De Paris Em 2012**. CONGRESSO MUNDIAL SOBRE RECURSOS EDUCACIONAIS ABERTOS (REA) DE 2012 UNESCO, Paris, 20 a 22 de junho de 2012. Disponível em: <[http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/WPFD2009/Portuguese\\_Declaration.html](http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/WPFD2009/Portuguese_Declaration.html)>. Acesso em: 10 de junho de 2015.

VAGLIATI, Ana Carla. **Gritos do Silêncio: o professor frente à violência sexual contra crianças e adolescentes no espaço escolar**. Dissertação (Mestrado em Educação) – Universidade Estadual do Oeste do Paraná – Campus de Francisco Beltrão, 2014, 176p.

ZIMERMAN, David E. **Etimologia de termos psicanalítico**. Porto Alegre: Artmed, 2012.