

UNIVERSIDADE FEDERAL DE MINAS GERAIS

EDUARDO AUGUSTO DA SILVA LEITE

Vídeo como serviço web

BELO HORIZONTE

2015

EDUARDO AUGUSTO DA SILVA LEITE

Vídeo como serviço web

Trabalho apresentado ao colegiado do curso de Cinema de Animação e Artes Digitais da Universidade Federal de Minas Gerais, como requisito parcial a obtenção do título de bacharel em Artes Visuais, Habilitação em Cinema de Animação.

Orientadora: Angélica Beatriz Castro Guimarães

BELO HORIZONTE

2015

SUMÁRIO

| | |
|---------------------------------------------------|----|
| 1.0 - Introdução..... | 3 |
| 2.0 - Tecnologias Web..... | 8 |
| 2.1 - Front-End..... | 8 |
| 2.1.1 - Hypertext Markup Language (HTML)..... | 8 |
| 2.1.2 - Cascading Style Sheets (CSS)..... | 9 |
| 2.1.3 - Javascript..... | 9 |
| 2.2 - Back-End..... | 10 |
| 3.0 - Histórico das Tecnologias Web..... | 11 |
| 3.1 - Década de 1990..... | 11 |
| 3.1.1 - Graphics Interchange Format (GIF)..... | 14 |
| 3.1.2 - Necessidade de Plugins..... | 14 |
| 3.2 - Anos 2000..... | 17 |
| 4.0 - Cinema Dinâmico na Web..... | 22 |
| 4.1 - Bases de dados e Narrativas..... | 23 |
| 4.2 - Feedback e evolução..... | 29 |
| 4.3 - Análise cultural e integração de dados..... | 30 |
| 4.4 - Futuro e Web 3.0..... | 32 |
| 5.0 - Conclusão..... | 34 |
| 6.0 - Glossário..... | 35 |
| 7.0 - Bibliografia..... | 37 |

1.0 - INTRODUÇÃO

A cada ano, a web vem influenciando nas interações sociais e econômicas das pessoas. Por se tratar de uma tecnologia aberta, universal e de crescente acesso, se abrem novas possibilidades de uso.

A web é um recurso derivado do suporte digital dos computadores, que estão conectados por toda a internet. Através da web é possível simular virtualmente todos os tipos de mídia conhecidos. A mídia analógica tradicional, quando é serializada¹ para o meio digital pode ser considerada uma **nova mídia** (MANOVICH, p. 49 2001).

Este trabalho tem como proposta descrever como a nova mídia, aplicada dentro do universo das tecnologias web possibilitam a existência de sites e os mais diversos tipos de aplicações. Em seguida, como elas se desenvolveram nas últimas décadas. Os modelos de negócios que foram se estabelecendo na web passariam a se apropriar da mídia digital de várias formas. Por fim, com as tecnologias web que se consolidaram nos últimos cinco anos (2010-2015) é proposto uma forma de integrar o cinema e seus processos através da automação da narrativa, dentro de uma aplicação web. A automação em questão seria baseada em utilizar um *software* para organizar as cenas, ações e atores organizados por uma base de dados.

O processo de automação digital na web, possibilitaria a reprodução de um filme onde a narrativa, personagens e cenários seriam estruturados em tempo de execução – no instante em que é requerido pelo usuário o carregamento de uma página web, e em resposta a suas interações. Essa automação seria variável de acordo com o usuário que utiliza a aplicação web, de forma semelhante como as pessoas consomem determinados serviços na web, como os de redes sociais.

O cinema de animação tradicional pode ser caracterizado pela exibição contínua de imagens e áudio durante um espaço limitado de tempo, através de um mecanismo que crie a ilusão de movimento. Sendo o mecanismo digital ou analógico, uma vez que a produção deste filme foi encerrada e ele foi armazenado

1 Conversão de objeto de mídia analógico em dados digitais.

para reprodução, ele não pode mais ser modificado durante o tempo em que é executado pela pessoa que o assiste, embora possa ser novamente modificado para constituir um novo filme. Podemos considerar que a maioria dos filmes consumidos pelas pessoas são lineares, por serem organizados a partir de um conhecimento humano que conta uma história a partir de uma sequência organizada de ações.

Dentro do suporte digital, a mídia é representada pela mesma linguagem que fundamenta os computadores. Assim a mídia é passível de ser re-interpretada por programas de computador. Em "*The Language of New Media*", Lev Manovich (2001, p. 49) estabelece cinco princípios para a **nova mídia**:

- **Representação matemática:** a arquitetura da nova mídia é representada matematicamente e executada por uma máquina digital.
- **Modularidade:** os componentes de uma mídia podem ser isolados em módulos independentes que atuam em conjunto para criar uma mídia diferente;
- **Automação:** os processos são realizados sem atividade humana e no exato momento em que são solicitados durante sua execução;
- **Variabilidade:** o contexto como são executados fornecerá parâmetros para que a aplicação entregue resultados diferentes, previstos na sua programação ou não;
- **Transcodificação:** os processos digitais resultarão em um objeto visual reconhecível por um humano. Exemplo: a imagem digitalizada de um texto legível a um humano através de uma interface gráfica, ou um filme exportado em arquivo de vídeo a partir de sua versão editável, para a máquina não passa de uma sequência binária de 0 e 1. O resultado de uma transcodificação de vários componentes lógicos, na tela de um computador no formato de uma imagem/interface reconhecível, também é conhecido como **renderização** ou *render*. (MEI, 2013)

Enquanto a mídia tradicional contém níveis de representação discreta, as amostras nunca são quantificadas. Essa quantificação de amostras são um passo crucial atingido pela digitalização. Mas por que, podemos perguntar, tecnologias de nova mídia são frequentemente em parte discretas? O conceito chave das semióticas modernas é que comunicação requer unidades discretas. Sem unidades discretas, não há linguagem. (MANOVICH, 2001, p. 49, tradução nossa)

Essa nova mídia pode ser uma representação digital de uma amostra analógica ou produzida de dentro do suporte digital. A forma como estamos acostumados a consumir a mídia fora do meio digital seria considerado a mídia tradicional ou analógica.

A exemplo do filme analógico, cada *frame*/fotograma ou “amostra” filmada e gravada em mídia fotográfica representa uma das unidades **contínuas** dessa mídia, onde não é possível para a máquina discriminar as informações matemáticas que a compõem. Essas amostras podem ser organizadas em sequência para compor o tempo de execução do filme. Como o tempo de execução é quantificável, essa propriedade do filme analógico passa a ser uma unidade **discreta**. A mídia tradicional pode ter tanto unidades contínuas como discretas.

Na nova mídia as amostras passam a ser representadas exclusivamente por unidades **discretas** da mídia, ao convertermos também os *frames* em *pixels* e metadados digitais². Quando as amostras do filme analógico são digitalizadas para um computador, todas as suas propriedades passam a ser quantificáveis por ele. Um exemplo seria a câmera digital, onde seus sensores recebem os raios de luz e os converte em informação matemática para dentro de um dispositivo de armazenamento (um cartão SD³). A informação captada posteriormente poderá ser visualizada ou editada em um software de imagem na forma de um arquivo. Já a câmera analógica grava a imagem diretamente em um material sensível a luz. A gravação é feita por um processo químico e torna mídia o próprio papel em que foi gravada. Ao digitalizar essa fotografia para o computador através de um scanner, seria possível ter acesso a essas informações que agora foram serializadas para o suporte digital.

2 Dados textuais que descrevem o objeto digital em questão

3 *Secure Digital Card* (SD) – unidade de armazenamento de dados utilizado em câmeras

Quando as amostras do filme analógico são digitalizadas, elas passam a ser representadas totalmente por unidades discretas. A quantidade de tons de vermelho, por exemplo, é uma unidade discreta contida em cada amostra do filme que pode agora ser lida ou alterada por um programa instantaneamente sem depender de processos de tratamento além do digital. Para uma mídia ser considerada nova, não basta apenas ter unidades discretas, essas unidades precisam estar digitalmente representadas e acessíveis por algoritmos⁴ capazes de as manipular. A capacidade de se quantificar digitalmente a mídia digital lhe atribui a característica de **representação matemática**.

Uma vez que as amostras, trilhas de áudio e demais informações estão devidamente digitalizadas, esses elementos podem ser separados em **módulos** ou bibliotecas de mídias digitais (*assets*), que podem ser replicadas, modificadas e reorganizadas dentro do computador para compor um novo filme. A partir de então, os módulos podem ser tratados diretamente pelo programa de computador **automaticamente**, sem um tratamento manual repetitivo por humanos. Como por exemplo, aplicando filtros de desfoque ou repetindo a execução da sequência de um filme várias vezes. A possibilidade de modularizar trechos do filme em objetos separados permite a utilização automatizada dos mesmos.

A quantidade de vezes que um trecho do filme será repetido, ou a intensidade do efeito de desfoque pode ser **variável**, decidido diretamente por um humano ou por uma máquina a partir de parâmetros colhidos em uma base de dados. A variabilidade auxilia a dar contexto ou parâmetros ao processo de automação utilizado. Todos os processos gerados pela máquina resultarão em **transcodificação** dos dados matemáticos. Os dados serão transcodificados em imagens ou outras informações culturais reconhecíveis por humanos – na interface final.

Os cinco princípios levantados por Manovich abrangem praticamente todas as formas de se trabalhar a mídia tradicional digitalizada no computador, para a produção de nova mídia. A web é uma tecnologia digital dependente de linguagens de programação, que exploram as propriedades discretas de um dado objeto e

4 Série de instruções em texto que possibilitam o computador a realizar tarefas programadas.

possui tecnologias que já atendem a diversos nichos de mídia digital. Nos últimos anos o crescimento de recursos disponíveis na web trouxe para o cinema de animação justamente sua natureza descentralizada e bem representada pelos cinco princípios.

A web é constituída de uma vasta quantidade de tecnologias, que desde os anos 1990 declinaram e ascenderam, conforme as demandas da indústria multibilhonária que se formou em torno da mesma. Uma forma simplificada de observar sua infra-estrutura pode ser feita sobre dois conceitos:

- **Cliente:** na web, o cliente é considerado um programa que interpreta um arquivo de dados fornecido por um servidor na internet e o converte em informação visual para o usuário, por exemplo os navegadores web como Mozilla Firefox ou Google Chrome. Todo o conteúdo gerado visualmente nos clientes/navegadores é de domínio das linguagens de programação *Front-end* – tem como “fim a frente”, onde as linguagens trabalharão a parte visível pelo usuário, a interface. O programa cliente pode solicitar arquivos de dados ao servidor através de *requests* ou **requisições**. Como em <http://exemplo.com.br/imagem.png>, é requerido do servidor um recurso de imagem localizado no dado endereço web ou URL.
- **Servidor:** os servidores são computadores físicos ou virtuais dedicados a retornar dados em resposta à requisição dos navegadores, de forma que os navegadores consigam transformar em interface gráfica para o usuário. Esses dados são comumente estruturados no servidor por alguma linguagem de programação *Back-end* (o “fim atrás”, tecnologias que trabalham abaixo do nível da interface) que esteja integrada em uma base de dados. (CAMARGO, 2008)

A web é constituída em sua base por vários servidores interligados pela internet. Os navegadores ou *browsers*, que se comunicam com esses servidores, retiram deles os dados necessários para desenhar/renderizar uma interface web funcional na tela do computador. Podem ser chamadas aplicações web os softwares que são executados em navegadores, capazes de gerar uma interface no lado

cliente a partir de um pré-processamento no lado servidor através das tecnologias *Front-end* e *Back-end*.

2.0 - TECNOLOGIAS WEB

2.1 - *Front-End*

O *Front-end* é a parte mais relevante ao cinema que busca criar trabalhos visuais compartilháveis na web. Os programas desenvolvidos para o navegador são interpretados em tempo real. Estes programas são conhecidos como *scripts*: o navegador lê o código escrito pelo programador assim que a página web é carregada e nela são executadas as instruções do programa. Esses códigos seguem padrões de formatação e marcação específicos para cada funcionalidade.

2.1.1 - *Hypertext Markup Language* (HTML)

O HTML é uma linguagem de marcação⁵ de hipertexto⁶, conceitualizada em 1989 por Tim Berners-Lee como um meio de distribuir conteúdo textual em rede. Uma linguagem de marcação não tem operadores lógicos⁷ necessários para ser considerada uma linguagem de programação. Sua função é delimitar partes do documento, como texto negrito ou uma imagem através de *tags* (WESLEY, 1998). As *tags* são estruturas textuais no arquivo HTML que demarcam uma área de texto a ser formatada quando exibida pelo navegador na forma de interface. Exemplo:

| Código HTML | Visualização no Navegador |
|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Um texto em <code>negrito</code> e um em <code>itálico</code> . | Um texto em negrito e em <i>itálico</i> . |

Neste caso, a *tag* `` (forte) realça o texto para o estilo de negrito, enquanto a *tag* `` (emphasis/ênfase) o torna em itálico. A delimitação de uma *tag* começa em `` e termina em ``. Essa área demarcada pela *tag* posteriormente pode servir de referência para que os *scripts* possam interagir com a página.

⁵ Linguagem que demarca áreas de texto a partir de uma sintaxe específica da linguagem

⁶ Texto capaz de interligar com informações contidas em outro local da web

⁷ Operadores que controlam o fluxo de execução de um programa, pré-requisito para ser uma linguagem de programação, ou uma linguagem "Turing Completa".

2.1.2 - *Cascading Style Sheets (CSS)*

O CSS é uma linguagem de folhas de estilo em cascata⁸ e, assim como o HTML, ela não é uma linguagem de programação, mas usa como base a marcação HTML para definir como será feita a apresentação visual da página, como cores de fundo e fonte do texto. Seu uso só ganhou mais importância quando passou a ser necessário separar a parte visual da marcação. Isso permitiu a produção de programas mais fáceis de serem mantidos e ampliados quase uma década depois da invenção do HTML (BOS, 1999). Com o CSS, o conteúdo da *tag* `` do exemplo anterior poderia ser colorido de vermelho ou o corpo de sua fonte ser 50% o tamanho da tela que o visualiza.

2.1.3 - Javascript

O Javascript é uma linguagem de programação. Sua função é tanto melhorar a interação da página com o usuário como também enviar requisições do navegador web para o servidor (técnica conhecida como Ajax) sem que o navegador precise carregar a página novamente. Com Javascript é possível criar jogos, animações e outros tipos de interações com o usuário que não seriam possíveis apenas com CSS/HTML. Por exemplo: quando o usuário clica em um elemento da página, pode-se iniciar uma animação de um elemento HTML movendo-o de um ponto a outro da tela, e ainda trocar sua cor a partir do que foi configurado no CSS. A manipulação desses elementos é feita através de um recurso chamado Javascript DOM (*Document Object Model*) (CHAMPEON, 2001). Em <http://bit.ly/1J4jrAn> realizo um trabalho interativo utilizando somente HTML/CSS/Javascript. São jogos simples para crianças, cuja interação é possível graças ao Javascript DOM.

Como quase sempre a construção de páginas em HTML envolve Javascript e CSS, geralmente o uso somente do termo HTML já inclui CSS e Javascript dentro dele como uma só tecnologia. Além do mais, Javascript não pode ser confundido com a linguagem Java, cuja maior semelhança é o nome que foi dado ao Javascript em razão da popularidade de Java para atrair desenvolvedores, além de uma sintaxe parecida.

⁸ Arquivos que contêm informação hierarquizada para a estilização de elementos HTML

Essas três linguagens são usadas por todos os navegadores web mais populares da atualidade, como Internet Explorer, Google Chrome e Mozilla Firefox (baseado no finado Netscape). Dentro desse contexto, alternativas a essas linguagens são implementadas pelos navegadores através de adaptações conhecidas como *plugins*. Os *plugins* são elementos especiais na marcação HTML que incorporam quadros com interfaces próprias de suas tecnologias, através das tags `<embed>` ou `<object>`. Os plugins mais utilizados são o **Flash Player** e o **Java**.

2.2 - *Back-End*

O *Back-end* é um ambiente mais diverso com relação às tecnologias utilizadas no lado servidor. O uso de cada uma varia tanto do programador quanto do problema que precisa ser resolvido no lado servidor. O *Back-end* é a área onde o cientista da computação será mais empregado, e para web são usados pelo menos dois níveis de aplicações. Eles comumente serão o software instalado (uma base de dados ou um conversor de vídeo) e a linguagem de programação que irá modelar as informações contidas no *Back-end* em dados compatíveis com as tecnologias do lado cliente.

Entre as linguagens *Back-end* mais populares para web estão PHP, Ruby (Rails), Javascript (Node.js⁹) e Python. Em destaque o PHP, trata-se da linguagem de programação mais utilizada para o desenvolvimento web, devido à facilidade de se criar programas em conjunto com o HTML. Plataformas importantes como Wikipédia `<http://pt.wikipedia.org>`, Facebook `<http://www.facebook.com>` e Wordpress `<http://www.wordpress.com>` foram construídas com PHP.

As linguagens citadas acima foram desenhadas principalmente para servir páginas na web. Mesmo cumprindo seu papel em tornar a web mais dinâmica, geralmente essas linguagens são mais lentas para resolver problemas que exigem muito processamento: converter um vídeo, acessar informações de dispositivos conectados diretamente na máquina ou gerenciar bases de dados, por exemplo. A

⁹ Node.js é baseado no V8, um interpretador Javascript contido em navegadores como Chrome que foi adaptado para a linguagem funcionar no lado servidor. Node.js pode ser considerado um “javascript de servidor”

vantagem do back-end é a capacidade de integrar com tecnologias criadas em outras linguagens instaladas no servidor, que possuem finalidades diferentes do que somente servir a web, incrementando a aplicação com novas funcionalidades. Exemplo de um serviço que precisa desse nível de otimização é o Youtube, para a conversão de inúmeros vídeos por minuto que linguagens como o PHP não possuem capacidade de resolver (e nem foram criadas para isso).

Um componente muito usado para dar o caráter dinâmico das páginas web são as bases de dados. Elas têm como função armazenar informações capturadas do usuário através da página web ou através de outros dispositivos com sensores e outras aplicações conectadas. O MySQL é uma das bases de dados mais usadas no mundo para servir páginas web. Parte do sucesso do MySQL está em sua integração nativa com o PHP, que simplifica para aqueles que estão iniciando o desenvolvimento web as relações entre páginas de internet e banco de dados.

3.0 - HISTÓRICO DAS TECNOLOGIAS WEB

3.1 - Década de 1990

A popularização da web como meio de comunicação na internet na década de 90 trouxe a um mundo frequentado principalmente por acadêmicos e companhias, usuários aptos a consumir serviços como e-mail, enciclopédias eletrônicas e portais de notícias. Não demoraria para que fossem oferecidos aos primeiros usuários leigos da internet, serviços de hospedagem de páginas web. Os *webmasters*¹⁰ tinham conhecimentos mínimos de programação (e menores ainda de design) e foram um dos primeiros “construtores” da web moderna fora do mundo corporativo e acadêmico. Até setembro de 1993, grande parte do acesso à internet era feito sazonalmente por estudantes de universidades. Provedores de internet como a AOL foram responsáveis pela popularização do acesso à web.

¹⁰ Administradores de sites

[...] Era o anual fluxo de Setembro de usuários leigos (*newbies*) que, sem qualquer senso de *netiquete*, faziam um completo estardalhaço. Isso coincidia com pessoas iniciando a universidade, conseguindo suas primeiras contas de internet, e fuçando sem se preocupar em aprender o aceitável. Esses esboços de novatos poderiam ser assimilados em alguns meses. Mas em Setembro de 1993, usuários da AOL também se tornaram capazes de postar na Usenet, praticamente ofuscando a capacidade dos ‘veteranos’ em assimilá-los; para aqueles que nostalgicamente se lembram dos velhos tempos, isso iniciou um inexorável declínio na qualidade das discussões em grupos de notícias. (THE JARGON FILE, tradução nossa)

Em alguns anos, a web passaria a ser dividida entre as pessoas que a constroem e aquelas que são capazes tão somente de interagir com ela. Embora o foco fosse na leitura, as páginas web não eram apenas um novo suporte para uma mídia de leitura como livros ou revistas de notícias. Ela era capaz de dar o poder aos usuários comuns a deixar sua marca sobre as primitivas ferramentas de comunicação social da época.

Enquanto mais e mais pessoas tinham computador e acesso à rede, cada vez menos estavam vendo o valor de suas contribuições. Como a maioria já consentia “não ter ideia sobre como usar computadores”, acabou virtualmente impossível a eles refletir no *medium* eles mesmos. Ao mesmo tempo, entretanto, esses *users*¹¹ usavam computador intensivamente, produzindo e fazendo upload de conteúdo sem parar. (LIALINA, 2009, tradução nossa)

Durante a década de 1990, um dos principais mecanismos da nova mídia era o hyperlink e uma interação mínima com funcionalidades dos *browsers* (navegadores) como scrollbars. Em <http://www.wwwww.jodi.org/> de 1995, por exemplo, pode ser observado o princípio modular no qual várias páginas independentes levam a mais informações conforme a pessoa navega. Um exemplo mais popular também poderia ser a Wikipedia que foi criada na década seguinte, mas um dos fatores que destacam o site jodi era como ele também representava a diferença entre o usuário leigo e os webmasters, que se mantém consistente até hoje. Para os desenvolvedores poderia ser um site comum cheio de detalhes escondidos no código-fonte¹², mas para o usuário leigo o site poderia ser um caso de conspiração governamental como relatos em <<http://bit.ly/1Gd5gC5>>, <<http://yhoo.it/1G1Vaov>> ou <<http://bit.ly/1G1VnYG>> criados por usuários comuns da web. Embora os leigos tivessem um espaço na web, até hoje atuam mais como

¹¹ *Losers* (perdedores) + *Users* (usuários), termo pejorativo para usuários leigos

¹² Código que será lido por um programa para desempenhar uma tarefa

produtores de conteúdo para veículos/plataformas de entretenimento e leitura na web, do que necessariamente participando da construção de sua arquitetura.

O conceito modular que o hiperlink oferecia também poderia ser abstraído como uma forma de contar histórias distribuindo páginas como cenas, e pequenas interações em Javascript para conduzir uma simples animação. A exemplo de <http://www.c3.hu/collection/agatha/>. Outro detalhe importante para uma narrativa é o som, que na época não haviam métodos simples de ser reproduzido em HTML, o exemplo citado precisava combinar HTML e Flash, ou outros plugins de multimídia como o Windows Media Player para conseguir áudio. Problema que foi solucionado apenas com a chegada do HTML5¹³ na década seguinte com a tag `<audio>`.

Com o crescimento na publicação de conteúdo na internet, em alguns anos o processo de automação e modularização de componentes das páginas web foram se tornando mais complexos para entregar informação de forma cada vez mais eficiente como indexadores de conteúdo¹⁴, *blogs*¹⁵, páginas de comentários e avaliações, *guestbooks*.

Esse é um dos meios no qual o princípio geral de variabilidade da nova mídia se manifesta. O objeto de nova mídia consiste de uma ou mais interfaces a uma base de dados de material multimídia. Se apenas uma interface é construída, o resultado será semelhante ao tradicional objeto de arte, mas isso é mais uma exceção do que norma. (MANOVICH, 1997, tradução nossa)

Embora a web ainda fosse primitiva para fornecer conteúdo em vídeo podemos afirmar que todo o conteúdo textual carregado em navegadores, conectados através de hiperlinks, são uma manifestação dos princípios de **variabilidade** e **automação** da nova mídia na web. Esses dois princípios são o que viabilizam todo o dinamismo da web, e a distanciam de formatos mais lineares de leitura como na mídia tradicional.

¹³ Versão 5 da linguagem HTML

¹⁴ Aplicações web que listam websites que possuem conteúdo relevante (*Google*)

¹⁵ Contração de *weblog*, que significa guardar informações na web em forma de diário

3.1.1 - *Graphics Interchange Format (GIF)*

Os principais formatos de imagens suportados no HTML, sem o uso de plugins, eram de três tipos: JPEG, PNG e GIF. Este último é um formato que permite ser animado e deixou um legado de animações e efeitos 3D pré-renderizados que dividiram opiniões. Sua portabilidade e simplicidade na produção de animações permitiu a distribuição em massa de imagens engraçadas e polêmicas que persiste até hoje. Também foi largamente utilizada como enfeite em sites. Os webmasters geralmente sem uma grande noção de design encontravam no GIF uma novidade em relação às páginas, muitas vezes estáticas, com alguns sites acabando poluídos por loops de mascotes dançando e glitter seguindo o cursor do mouse. Em <http://art.teleportacia.org/exhibition/GRAVITY/>, de 2003, é possível observar a interação limitada que o HTML fornecia, onde o foguete animado em GIF decola ou retorna à Terra a partir da interação com o controle da barra de rolagem comum nas páginas web.

O GIF para a animação possuía muitas limitações: não executava áudio nem era interativo, além de reproduzir uma quantidade baixa de cores (256). Sozinho, o GIF, ainda estava mais para os antigos filmes mudos do cinema tradicional, só que com mais algumas cores. A falta de uma solução para reproduzir animações com maior qualidade na web impulsionou o uso de tecnologias como o Flash¹⁶.

3.1.2 - *Necessidade de Plugins*

Inicialmente o HTML era dedicado a distribuir documentos de texto simples com imagens na internet. Com o tempo, passou a ser visto como uma mídia importante que traria o mundo das aplicações distribuídas na web, ao invés do modelo vigente de instalar softwares diferentes em um computador. Na época, a evolução dos padrões web e a adoção deles pelos desenvolvedores eram muito lentas. Poderia ser devido às limitações e divergências nas implementações do HTML nos navegadores, predominantemente proprietários, como Internet Explorer e Netscape. A demanda por alternativas que buscariam unificar a criação de conteúdo visual independente do fabricante do navegador, passou a crescer.

¹⁶ Tecnologia voltada para animação vetorial na web

Essa carência impulsionou a popularidade do plugin Flash Player, então desenvolvido pela Macromedia, com a promessa de enriquecer a experiência com a web enquanto a Sun Microsystems lançara o plugin web para a linguagem Java.

Os plugins não eram uma parte integral do HTML, tanto Java quanto Flash exigiam conhecer uma nova linguagem de programação para ambos. Eram basicamente quadros embutidos dentro de um documento HTML através da tag `<embed>` ou `<object>` e o seu conteúdo era exibido como animações dentro deles através de aplicações conhecidas como máquinas virtuais¹⁷, desenvolvidos para cada sistema operacional. Devido à complexidade da linguagem Java e por consumir mais memória que seu concorrente, o Flash – que além de tudo tinha uma interface visual para a produção de animações – foi ganhando cada vez mais espaço entre animadores e designers. O reinado do Flash como padrão de conteúdo rico na web foi instalado com sucesso.

No Brasil a animação na internet nessa época era feita principalmente através de canais como humortadela.com.br, charges.uol.com.br e mundocanibal.com.br, que se tornaram muito populares, principalmente para o público infanto-juvenil. Nenhum desses três sites (e muitos outros pelo mundo afora) seriam viáveis se não fosse pela versatilidade do Flash. O plugin possibilitava imagens vetoriais que eram mais fáceis de trafegar na conexão de internet que o *bitmap*¹⁸, produzidas através de uma ferramenta intuitiva e simples. Todos os formatos web abertos¹⁹ até então eram bitmaps e muito aquém de possibilitarem animações bem construídas. O domínio do Flash continuará até uma maior diversificação dos dispositivos que exibem conteúdo da web e o aumento da velocidade média de conexão das pessoas pelo mundo.

Uma característica importante do Flash, a respeito da automação e variabilidade na nova mídia era sua capacidade de realizar requisições a um servidor durante a execução do filme. Isso possibilitaria adicionar novos clipes de filme durante a execução do vídeo principal ao se comunicar com o servidor novamente. A exemplo de alguns trabalhos de **David Clark** como Ulysses, no site Chemical Pictures (<http://bit.ly/1LfFSA6>), o conceito de filme interativo é seguido

¹⁷ Softwares que facilitam a execução de uma determinada linguagem em múltiplas plataformas

¹⁸ Formato de arquivo que armazena cada informação de *pixel* da imagem

¹⁹ Linguagens usadas na web que são *Open-Source*

onde na página HTML incorpora o quadro do plugin Flash, e a interação do filme fará novas requisições que incluirão outras combinações de trechos de filme. O fundamento de variabilidade é respeitado pelo fato da reprodução do filme ocorrer de forma discreta conforme há interação.

A principal resistência que o HTML fornecia ao Flash nessa época era sua sintaxe de programação simplificada, por ser simples construir interfaces com a linguagem de marcação. Além disso havia a facilidade de integrar com linguagens *Back-end* para exibir informações dinâmicas retiradas de bases de dados. Então o HTML se manteve consistente no terreno de tecnologia de informação e por muitos webdesigners, enquanto o Flash era usado predominantemente por animadores. Esse cenário de dualidade mudaria a partir dos anos 2000.

Enquanto o HTML e o Flash eram principalmente disputados por artistas, historicamente o plugin Java tinha um nicho mais voltado aos programadores de carreira. A linguagem Java é até hoje uma das linguagens mais populares do mundo, devido a sua consistência em várias plataformas²⁰ diferentes e seu design melhorado que carecia a linguagem na qual foi inspirada, C++. Como Java já era uma linguagem amplamente usada no desenvolvimento em vários setores, o plugin web semelhante ao funcionamento do Flash foi criado como uma nova porta para esta linguagem. Apesar de sua complexidade para um artista, ela não deixou de ser usada por aqueles que tinham afinidade com programação mais complexa como no caso de bibliotecas como o **processing**.²¹

Em <http://www.ljudmila.org/~vuk/ascii/film/> do artista **Vuk Cosic**, os vários frames de filmes antigos foram convertidos em caracteres especiais dentro do padrão ASCII. Este trabalho pode demonstrar a maior vantagem de Java frente ao HTML/Flash: é uma linguagem bem consolidada e popular antes da chegada do seu próprio plugin. Logo não faltaram bibliotecas de códigos²² para auxiliar no processamento desses filmes que em Flash seria difícil, e em HTML até então, impossível.

²⁰ Sistema operacional e tipo de dispositivo que suporta determinada tecnologia.

²¹ Tecnologia em Java utilizada por artistas para produzir arte computacional.

²² Algoritmos criados por outros desenvolvedores que podem ser re- aproveitados em novo código.

Enquanto aplicações em Flash e HTML são concebidas a partir de um critério visual e são programáveis, Java tem uma visão mais científica. Essa visão permitiu ao artista explorar com mais detalhes as unidades discretas contidas nos filmes digitalizados, como os gradientes de cinza, e os transformar em caracteres. Conforme o HTML e Flash evoluíram, eles foram incorporando cada vez mais essas características. Por fim, como Java já era usado em *Back-end*, o plugin oferecia uma forma de se desenvolver para web usando apenas uma linguagem.

3.2 - Anos 2000

Com a entrada de novos dispositivos no mercado, a necessidade de investir no código aberto (*open-source*) tornou-se mais evidente. As tecnologias abertas poderiam ser desenvolvidas de forma descentralizada em várias plataformas por desenvolvedores independentes, sem um desenvolvedor central com as patentes para o licenciamento de uma determinada tecnologia.

Também até os anos 2000 as páginas tinham como foco um conteúdo mais informativo e estático, e boa parte das aplicações de grande porte ainda eram executadas em ambientes Desktop, dependentes de serem instalados em cada máquina para serem utilizados. A mudança significativa está em como os negócios passaram a ser feitos com a web após essa época: muitas aplicações foram migradas como serviços dedicados na web (como plataforma), que poderiam ser acessados a qualquer momento pela web. Esse modelo possibilitou como nunca antes a obtenção massiva de dados fornecidos pelos usuários que as utilizavam, permitindo aos desenvolvedores obterem estatísticas de uso de seus produtos e conseguirem prever melhorias e expansões para os mesmos. Essa pressão das novas tecnologias e plataformas colocaria em evidência a necessidade de melhorar os recursos do HTML frente ao Flash Player, que era propriedade da Adobe após a compra da Macromedia, ou seja: o seu avanço estava totalmente dependente das intenções da empresa com essa tecnologia, retirando o poder de melhorar a sua implementação através de outros grupos de desenvolvedores.

Em 2004 o HTML (já em sua versão 4) passou a chamar a atenção de outros grupos de desenvolvedores que acreditavam que o padrão poderia ser expandido e

receber novas funcionalidades dentro de um navegador. A iniciativa foi tomada pela Mozilla, Apple e Opera. Mas inicialmente foi rejeitada pela W3C (W3C, 2014)²³, que estava concentrada num substituto do HTML baseado em XML - o XHTML.

Em contrapartida à rejeição da W3C, os três grupos desenvolvedores formaram a WHATWG²⁴ que propôs o HTML5 como futura versão. Esses avanços, em conjunto com mudanças na disponibilização de serviços na web, trouxeram uma percepção de que a web estava passando por uma atualização, conhecida como Web 2.0 discutida pela primeira vez por Tim O'Reilly na Web 2.0 Summit também em 2004.

O conceito, era baseado na seguinte premissa:

A Web 2.0 é a rede como plataforma, cobrindo todos os aparelhos conectados; Aplicações web 2.0 estão entregando software como um serviço continuamente atualizado que melhora conforme mais pessoas a usam, consumindo e combinando dados de múltiplas fontes, incluindo usuários individuais, enquanto fornecem seus próprios dados e serviços em um formato que permite ser re-combinado por outros, criando uma cadeia de efeitos através da “arquitetura da participação” e entregando experiências de usuário mais ricas. (O'REILLY, 2005)

Com a popularização simultânea do HTML5 e o termo Web 2.0 que ainda estava em construção, muitos produtos que disponibilizassem recursos avançados de Front-end eram vendidos como Web 2.0, enquanto a Web 2.0 tratava na verdade de uma mudança na interação conjunta do *Front-end* com o *Back-end*. Todo o tratamento de dados cedidos pelo usuário que caracterizava o dinamismo dessas plataformas só podiam ser realizados no lado servidor. Muitas das tecnologias utilizadas na Web 2.0 já eram existentes desde a década de 1990: a Web 2.0 trata mais de um amadurecimento do modelo entre cliente/servidor do que necessariamente novas tecnologias.

Embora o *Back-end* seja de domínio prioritário do programador, é importante conhecê-lo em conjunto com o modelo de negócios estabelecido pela Web 2.0 para

23 W3C - Sigla para World Wide Web Consortium, entidade responsável pela padronização e organização de tecnologias web, principalmente a especificação do HTML. Foi criada em 1994 por Tim Berners-Lee.

24 WHATWG - Sigla para Web Hypertext Application Technology Working Group.

viabilizar um trabalho artístico capaz de atuar como uma plataforma, e trazer o modelo de nova mídia para um novo nível de interatividade.

Com o passar do tempo o HTML5 passou a substituir o Flash Player em diversas aplicações para web embora ainda no contexto da animação, o fato do Flash possuir uma interface amigável para o desenvolvimento de filmes ainda o torna uma ferramenta de uso superior para tal finalidade. Mas, como o HTML5 é uma plataforma aberta e sempre em construção, a tendência é essa diferença entre funcionalidades diminuir, visto que só tem ganhado espaço em todas as plataformas Mobile (*Tablets* e *Smartphones*) e Desktop (computadores de mesa ou *Notebooks*) e vem sendo implementada também em TVs e executáveis nativos (que não precisam ser acessados através da internet, mas instalados como um programa tradicional no computador).

Além dos recursos já citados, o padrão HTML5 segue com novas funcionalidades. Entre as relevantes para o campo da animação estão a possibilidade de temporizar animações, criar desenhos tridimensionais e bidimensionais, gerar imagens em vetor ou *bitmap* e eventos de toque. Outros recursos importantes são:

- Geolocalização, acesso às coordenadas (latitude/longitude) de quem usa a aplicação;
- Integração com periféricos como *joysticks*, microfone e câmera;
- *Websockets*: comunicação constante e de duas vias entre cliente/servidor;
- Vídeo e Áudio, sem a necessidade de *plugins* de terceiros;

Se a discussão estivesse restrita tão somente na produção de filmes tradicionais, provavelmente o Flash ainda seria o vencedor. Embora o HTML5 forneça uma curva de aprendizado maior para a criação de animações, traz consigo uma grande variedade de recursos e flexibilidade com diversos tipos de dispositivos e resoluções de tela²⁵. Isso permite buscar novos métodos não só visuais, mas ligados ao roteiro sem falar, necessariamente, de um jogo onde a condução da narrativa é feita pelo jogador conforme ele toma escolhas.

²⁵ Quantidade de *pixels* dispostos na tela

Companhias com serviços Web 2.0 dependem da interpretação de dados fornecidos por quem usa suas plataformas em produtos cada vez mais eficientes. Hoje um avião envia para um servidor de centrais aéreas mais de 1TB²⁶ de dados com informações de navegação e posição do veículo, permitindo facilmente uma leitura detalhada da viagem e, se necessário, uma simulação fiel do que fora realizado, uma quantidade de dados impossível de ser analisada somente por humanos. (THOMSON, 2010). Com a ascensão de tecnologias como o Big Data, as máquinas com capacidade de aprendizado exercem um importante papel na interpretação dessas informações. Embora, por motivos legais, dados de tráfego de aviões estejam restritos às companhias de vôo e governo, o mesmo tipo de captação é feito o tempo inteiro em redes sociais. O Facebook por exemplo, é capaz de prever alterações no relacionamento de pessoas antes que aconteçam. A Amazon pode prever a iminência de uma compra e veicular o produto ao galpão mais próximo da pessoa antes que ela confirme o pedido, a fim de reduzir o tempo de espera (LOMAS, 2014). Nos locais mais obscuros dos laboratórios do Google, um usuário pode ter entregue tantos dados particulares através de produtos como o buscador, o Google Plus, Gmail, Google Chrome e similares que eles já possuem informações o suficiente para simular parte da interatividade de uma pessoa na web e executar experiências para novos produtos que se enquadrariam a pessoas que possuem um mesmo padrão de utilização desses produtos (LEONHARD, 2014).

Para se examinar a importância dessa captação de dados, realizada todos os dias para o modelo de negócios que a web 2.0 representa, analisemos o caso do Youtube <<http://www.youtube.com>>: é uma vasta base que capta por minuto aproximadamente 300 horas de vídeo <<http://bit.ly/PH4VFv>>. Como essa grande base de dados poderia intensificar o processo artístico na web?

Muitos serviços de redes sociais perceberam que expor sua base de dados para uso do público ajudaria não só a consolidar sua própria plataforma, como a criar serviços terceiros dependentes dela, o nome dado a essas bases públicas é API (*Application Programming Interface*) e as aplicações que combinam essas várias APIs para propor novas funcionalidades são consideradas **mashups**. Esse conceito já era utilizado no nível dos sistemas operacionais como o Windows para padronizar

26 1 Terabyte: 1024 Gigabytes

o desenvolvimento de software para aquele sistema. Ele acabou sendo adaptado para a web através de convenções na forma como os aplicativos terceiros fazem requisições à plataforma.

No caso do artista **Vuk Cosic** podemos fazer por exemplo uma releitura de seu trabalho considerando que durante a “Web1.0” o seu trabalho ainda estava restrito ao próprio servidor onde estava publicado. Em <http://dsleite.com.br/tcc/conversor> recreei parte do processo de se fazer uma leitura dos valores discretos de um vídeo e os converto, cada *frame*, em um conjunto de caracteres assim como o original, só que dentro de um documento HTML e não do *plugin* Java. A diferença é que agora a exibição do vídeo não está mais restrita àqueles que o autor, nos primórdios da web, escolheu disponibilizar em seu site em seu próprio processo de representar como seriam os primeiros vídeos da web, em analogia ao início do cinema. Esse mesmo processo agora torna-se mais amplo por suportar qualquer um dos milhares de vídeos disponíveis na base de dados do Youtube graças às APIs de dados. O *Back-end* faz o seu trabalho de se comunicar com os servidores do Youtube, baixa o vídeo e o converte em texto, o *Front-end* reproduz cada *frame* convertido, um após o outro, criando a ilusão de movimento, que na década de 1990 era inviável de ser realizado por HTML sem um grande prejuízo na memória do computador.

Pode-se considerar que agora a nova mídia não está restrita apenas ao objeto digital que pode ser alterado diretamente pelo usuário, mas ela também se modifica além do espaço geográfico através de bases de dados dinâmicas ao redor do globo que podem “conversar entre si”.

4.0 - CINEMA DINÂMICO NA WEB

Com o processador de vídeos do Youtube descrito anteriormente, o procedimento de acesso à base de dados da rede social e a conversão dos vídeos em texto só foi possível ao se atender duas condições: uma consulta textual, e a seleção do vídeo a ser convertido. Essas condições que são executadas pelo usuário são parâmetros para que toda a estrutura automatizada da aplicação web seja capaz de desempenhar a sua função de baixar vídeos da rede social e os converter em uma nova representação visual.

Do ponto de vista dos processos da produção de uma animação, a área de atuação da aplicação descrita anteriormente, por editar um vídeo já produzido, estaria relacionada ao processo conhecido como pós-produção. As propriedades discretas dos *frames* do vídeo são reinterpretadas pelo *Back-end* retornando uma nova organização do *frame* com uma aparência diferente.

Se retornarmos ao Ulysses de David Clark podemos visualizar um processo semelhante de pós produção, mas referente a montagem das cenas: três quadros abertos com trechos de vídeos aleatórios para que o usuário da página possa trocar até ficar na ordem que ele quiser. No caso de um cinema dinâmico, essa interação deixaria um registro na base de dados do servidor de cada escolha do usuário a partir da interface web, e posteriormente utilizadas pela própria máquina para organizar suas próprias cenas de forma mais precisa no futuro.

A capacidade de automação que as máquinas possuem hoje poderia abranger os vários processos do desenvolvimento de um filme. Informações passadas pelo usuário, ou acumuladas por outras máquinas para a base de dados, tornaria possível a máquina tornar mais diversa a sua produção audiovisual. A importância da web para esse tipo de procedimento, é que pela sua facilidade de acesso por pessoas de todo o mundo, ela funciona como um grande sensor para a captação de informações, através de sua interface capaz de captar interação do usuário.

Embora um computador não tenha capacidade de visualizar se a sua produção é eficiente ou de bom gosto, ela está dependente daquele que a assiste e gera *feedback* (retorna para a máquina se o resultado foi bom ou não). De certa forma, não tão diferente do que já ocorre na atualidade, onde seriados e franquias inteiras podem ser cancelados caso não obtenham sucesso imediato. A diferença é que um “roteirista” Web 2.0 seria capaz de construir, assim que requisitado, a produção de um filme, alternando entre atores, cenas e ações. A velocidade com que essa aplicação pode construir filmes mais próximos das escolhas ou perfis de seu público dependerá da quantidade de *feedbacks* que ela receber dos usuários.

A aplicação web passa a ter como seu ponto estratégico agora as informações de sua base de dados. Ao transformar o usuário naquele que enriquece as funcionalidades do próprio produto, ela passa a funcionar como uma plataforma dedicada na web, dentro dos paradigmas da Web 2.0.

4.1 - Bases de dados e Narrativas

O usuário da narrativa está atravessando a base de dados, seguindo conexões entre seus registros como estabelecidos pelo criador da base de dados. Uma narrativa interativa (que também pode ser chamada de hiper-narrativa, em uma analogia com hiper-texto) pode então ser entendida como a soma de múltiplas trajetórias pela base de dados. Uma narrativa linear é uma, entre muitas outras possíveis trajetórias: por exemplo uma escolha feita dentro da hiper-narrativa. (MANOVICH, 1997, tradução nossa)

Tanto o conversor de vídeo quanto Ulysses ainda estão presos ao final do desenvolvimento de um filme. Para que possamos processar um filme desde o início de sua criação precisamos também converter o pensamento textual/intelectual, ou roteiro, em representação matemática, para que possa ser analisado por um algoritmo. Ao se interligar dentro de uma estrutura de base de dados os seguintes elementos:

- Elementos narrativos (roteiro);
- Elementos visuais (filmagens de cenas, atores);
- Sensores (capturas da interação do usuário – *Back-end*);
- Interface (uma página HTML – *Front-end*)

Conseguimos criar uma integração entre elementos existentes em uma determinada narrativa, e a capacidade de organizá-los de várias formas diferentes dentro da plataforma digital.

O site WaxWeb <http://www.waxweb.org/>, por David Blair, reivindica o título de “o primeiro filme online”. Nele é executado um filme onde as cenas possuem um hyperlink contendo informações relacionadas à ela. Por exemplo, o texto da narração em *off* (é narrado por um personagem externo fora da cena) que está ocorrendo naquela cena, e comentários do diretor estão disponíveis nos *hyperlinks*. O filme pode ser assistido linearmente, mas esse fluxo pode ser rompido, se o usuário decidir navegar pelos *hyperlinks* para enriquecer a narrativa. Embora seja um exemplo de cinema dinâmico, o filme se mantém inalterado.

As bases de dados podem organizar informações de modo caótico, sem a linearidade de um texto narrativo. O resgate dos dados contidos na base, em um formato organizado, só pode ser feita através de um programa que é designado para extrair dados relevantes para seu próprio funcionamento.

Em computação, uma base de dados é um local na máquina que armazena dados para posterior consulta. Tanto o texto de Manovich (*The Language of New Media*) como os trabalhos de web-arte citados até o momento foram concebidos antes mesmo dos primeiros artigos que descreveriam o modelo da Web 2.0, em 2005. Até então o acesso à base de dados na web era majoritariamente qualquer documento digital armazenado no computador que pudesse ser alcançado pelo usuário através de um *hyperlink* ou como *requisição*.

David Clark e Blair desenvolveram trabalhos dentro desse perfil “pré Web 2.0” de acesso à base por também estarem inseridos na década de 1990. Isso é resultado do próprio conceito de se construir páginas web na época. As páginas tinham pouca dependência em sistemas dedicados em gerenciar dados que passaram a sustentar o modelo da Web 2.0. Mas existiam em pequenas escalas como guestbooks e seções de comentários.

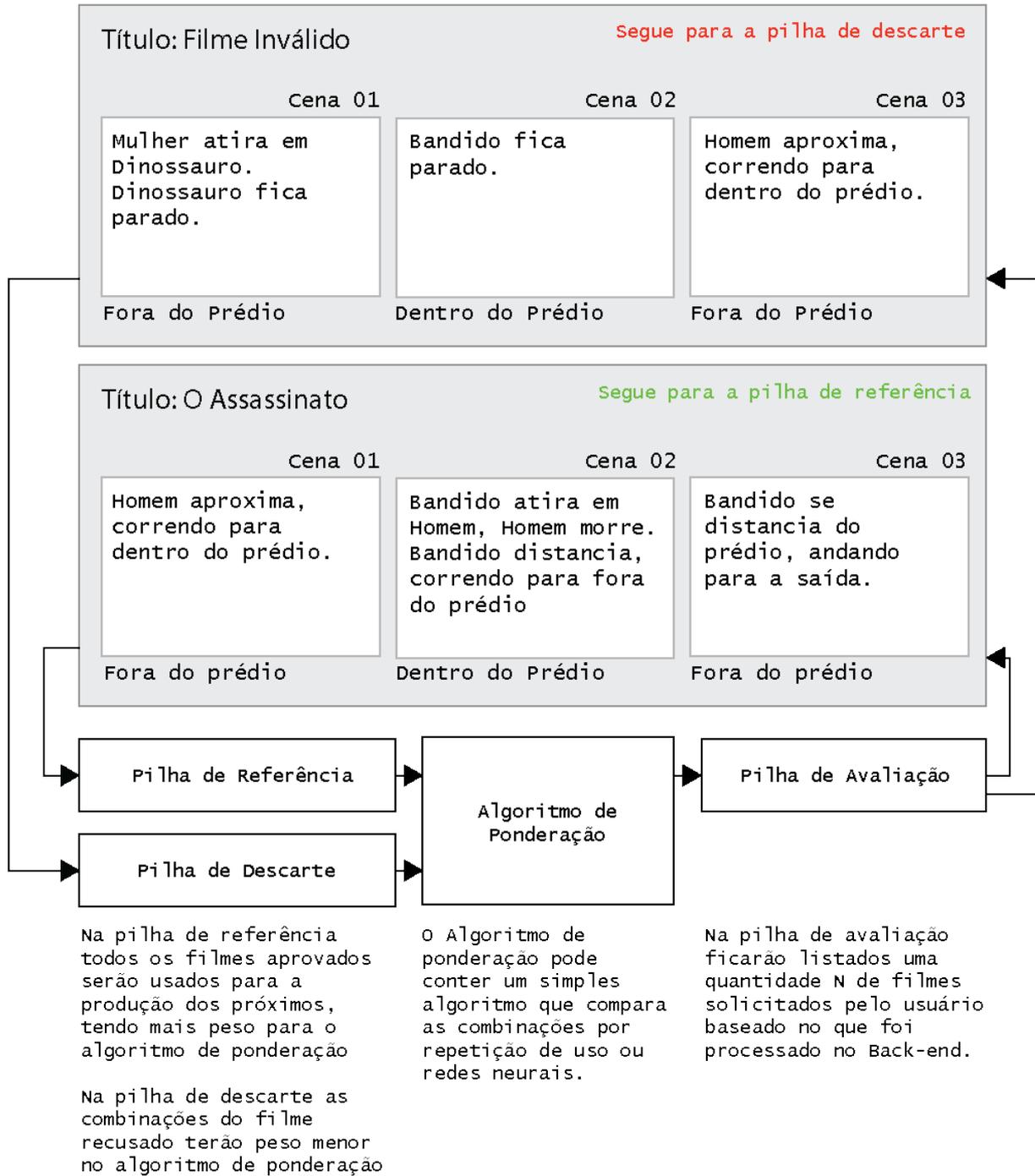
A desvantagem do modelo pré Web 2.0 é que ele está muitas vezes misturado com a estrutura de dados que tratará também da apresentação, e os links para cada local precisarão ser editados manualmente pelo criador da página. Essa mistura dificulta que o usuário externo influencie a base de dados de forma segura, já que é danoso para o próprio website a alteração de dados por terceiros no mesmo local onde a interface é armazenada. Embora seja possível um trabalho em que se possa inter-conectar outras mídias e interfaces com cenas do filme, a interferência humana pode aumentar as chances de erros de programação e brechas de segurança na aplicação.

A base dados na Web 2.0 está comprometida em isolar seu conteúdo da informação exibida ao usuário através de padrões e softwares dedicados. Assim criando um ambiente centralizado, sem ter seu código mesclado à página web. Isso permite que a base de dados exerça funcionalidades exclusivas à otimização de seu conteúdo já que os dados são processados no servidor antes de serem apresentados junto com a interface web. Também é mais seguro filtrar e modelar quaisquer informações captadas do usuário. Podemos dividir a organização de dados em dois modelos:

- **Arquivos de texto** - Planilhas, XML, texto CSV, VRML, COLLADA, arquivo de texto comum (.txt).
- **SGBDs** – Aplicações dedicadas a administrar dados: Relacionais (MySQL/Postgres), Documento (MongoDB), Gráficas (Neo4j/OrientDB).

Considerando uma narrativa em que o roteiro do filme é dominado por ações, e essas ações são realizadas por *atores* que estão contidos em *cenas*. Cada ator, ação e cena podem ser reorganizados dentro de um espaço de tempo, constituindo a narrativa de um filme de alguns segundos. A cada execução do usuário, o filme será reorganizado de forma diferente. A seguir, como poderia funcionar a lógica:

O Filme chega randômico do Back-end para avaliação na interface Front-end



Haveriam pelo menos quatro personagens. Poderiam aparecer desde 1 a 3 personagens diferentes ou repetidos numa mesma cena.

| | |
|-------------------|----------------------------------------------------------|
| Homem | <i>Um personagem genérico representando um homem.</i> |
| Mulher | <i>Um personagem genérico representando uma mulher.</i> |
| Bandido | <i>Um personagem com roupa listrada de presidiário.</i> |
| Dinossauro | <i>Um personagem humanóide com cabeça de dinossauro.</i> |

Lista de ações. Cada personagem pode ter até 4 ações seguidas, o total será decidido pela máquina. A execução de todas as ações finalizam a cena.

| | |
|-----------------------------------------|------------------------------------------|
| <i>Sem ação</i> | <i>É Golpeado/Morre</i> |
| <i>Ataca {personagem} com {arma}</i> | <i>Aproxima/Distancia, correndo para</i> |
| <i>Aproxima/Distancia, andando para</i> | <i>{personagem} abraça {personagem}</i> |

Se considera que o filme teria uma combinação de três cenas seguidas, com três de quatro personagens em cada cena, com até quatro ações por cena, uma para cada personagem e dois cenários. O programa contaria com milhões de estados (quantidade combinações diferentes) que resultarão em um filme de três cenas, com a duração de alguns segundos. Dessa variedade, a maioria das combinações resultarão em cenas sem nenhum significado e parte do significado será carregado pela aparência física do personagem nas variadas condições do programa. Por fim, fica em aberto a nomeação do título do filme, que facilita o entendimento do que irá ocorrer.

Com a classificação dos tipos de dados em cenários, personagens e ações, o processo de geração do roteiro já pode ocorrer no *Back-end*. Até este ponto, a aplicação já desempenha um comportamento semelhante ao de aplicações Web 2.0 pois já absorve dados fornecidos por usuários e tenta dar algum sentido aos dados para aprimorar o desempenho da própria aplicação. Agora o *Back-end* precisa ser exposto para a interação com o usuário. Assim como redes sociais como o Facebook e Youtube, seria exposta uma API de dados para que comandos possam ser enviados do *Front-end* (ou interface HTML). O aspecto da API é importante nesse momento, pois é ela que irá transcodificar todo o “conhecimento” acumulado no *Back-end* para uma interface visual no *Front-End*. Com uma API pública exposta na web, qualquer aplicação que implemente os personagens, cenários e ações para comunicação criaria um filme com estética diferente.

Como exemplo de variabilidade estética, pessoas em qualquer lugar do mundo poderiam produzir filmes em *stop-motion*²⁷, *live-action*²⁸, 3D e 2D desde que criando todos os objetos e animações suportados pela plataforma. Esse método migraria para a “nuvem” partes do processo de produção de um filme onde a produção visual teria suporte de um “conhecimento acumulado” posterior na parte de narrativa.

Vale lembrar que a capacidade criativa das máquinas é muito restrita, pois computadores só podem tomar decisões ou realizar tarefas a partir de unidades discretas. No nosso atual estado tecnológico, os computadores só conseguem desempenhar suas tarefas sobre dados pré-digitalizados do ambiente externo. Uma vez que as máquinas carecem de capacidade de dar sentido aos dados a ponto de produzir novo conhecimento como os humanos.

Na proposta de filme dinâmico, não importa para o sistema planejar cenas e ações que provoquem um determinado impacto ao usuário. Um personagem matar o outro não passa apenas de uma decisão estocástica, viciada pelos dados de referência fornecidos pelos usuários. No futuro, da mesma forma que o sistema expõe uma API de dados para a integração das interfaces visuais e suas estéticas, nada impede que exista ainda outra API criada por terceiros capaz de somente interpretar o que as ações querem dizer. Dessa forma criando vários níveis de dependência de máquinas, que aprendem em conjunto pela web sem interação humana.

Um possível exemplo de API de terceiros que poderia implementar sentido entre as ações, está o *Moral Storytelling System* (MOSS).

Um autor humano simplesmente decide um caminho emocional para a história, e o computador faz o resto. O computador seleciona os eventos que evidenciam essas respostas emocionais dos personagens, a trama automatizará as ações necessárias para cada personagem. (REILLY, 2014, tradução nossa)

²⁷ Animação onde são fotografadas as poses de um personagem para a criação de movimento.

²⁸ Filmagem com atores reais.

A relevância do sistema poder discernir a moralidade entre as ações dos personagens é que o fator randômico e a dependência de uma decisão humana é reduzida.

Construir uma narrativa não se trata apenas de organizar informações de forma que façam sentido, neste caso um humano pré-seleciona o tipo de emoção que a fábula gerada automaticamente tentará elucidar. E mesmo assim, só humanos são capazes de desenvolver histórias simples ou complexas sem pensar demais, enquanto computadores precisam de uma quantidade massiva de instruções que ainda está aquém da capacidade humana.

4.2 - Feedback e evolução

Deve-se ao conexionismo, entre outras coisas, a invenção das “redes neurais”, redes virtuais calculadas pelo computador que simulam células vivas e produzem, pela maneira como elas estão interconectadas, comportamentos que nenhuma delas, tomadas individualmente, poderia assumir. (COUCHOT, p. 27, 2003)

Um dos primeiros problemas a serem encontrados, uma vez que a máquina é capaz de sortear e compor um filme inteiro randomicamente, é que este fator não denota ação humana no que diz respeito a construir um filme com o mínimo de sentido.

Quando se cria uma narrativa para o cinema, estabelecer a ordem linear de cenas e ações trará um tipo de ideia a ser entendida por quem assiste o filme. Neste caso, a base de dados rompe com a linearidade expondo o caos de informações sem contexto. Os algoritmos no *Back-end* que fazem comunicação com a base de dados serão responsáveis por conectar os dados e retornar ao *Front-end* como a interface será estruturada para apresentar o filme.

Na ausência de um roteirista humano para organizar esses dados, e a incapacidade da máquina resolver esse problema randomicamente com eficiência, o usuário passa a ser a principal origem dos parâmetros que ajudarão a máquina a produzir o filme. Chegando a algo próximo do modelo de negócios da web 2.0, que concentra informações de usuários para processamento, o que transformou simples páginas Web da década de 1990 em grandes plataformas sociais.

Através de uma interface, o usuário indicaria a melhor combinação de cenas na sua preferência durante a visualização de um filme randômico. As mais frequentes combinações de cenas preferidas pelo usuário, seriam armazenadas na base de dados para futuro processamento. A partir daí, o algoritmo no *Back-end* composto de redes neurais e base de dados seria capaz de equilibrar as escolhas e criar filmes mais próximos, em média, do que foi escolhido por quem acessou a aplicação.

Já existem ferramentas no mercado para automação de animação na web como <http://goanimate.com/> e <http://www.wideo.co/>. A diferença fundamental entre esses dois produtos e a proposta desenvolvida neste trabalho, é da proposta buscar automatizar inclusive a necessidade de alguém organizar ideias num filme e seu usuário ser responsável somente por verificar se aquela produção está dentro do que busca assistir.

4.3 - Análise cultural e integração de dados

Mais precisamente, uma base de dados pode suportar narrativa, mas não há nada na própria lógica dela que a alimentaria. Sem surpresa, portanto, que bases de dados ocupam um significativo, senão o maior, território no espectro da nova mídia. O mais intrigante é que o outro fim do espectro - narrativas - continuam a existir na nova mídia. (MANOVICH, 1997)

Ao separarmos todos os elementos posteriores à própria organização da narrativa em uma base de dados, o roteiro passa a ser somente um dos derivados dela, já que as informações distribuídas de forma caótica na base de dados, foram organizadas de forma a gerar um história através da linguagem de *Back-end*. Criando aplicações com diferentes propósitos, há a possibilidade de se construir novas interfaces que não uma narrativa, com esses mesmos dados.

Expandindo a quantidade de ações, personagens, cenas e adicionando capacidade de captar informações como faixa etária, localização e demais dados pessoais, tornaria viável utilizar esse tipo de aplicação para avaliar tendências em vários grupos de usuários, podendo antecipar ou direcionar a produção de filmes para um determinado público do ponto de vista estatístico.

Nosso diretor executivo gosta de dizer que o antigo modelo é escrever um artigo e rezar para que um milhão de pessoas o leia, enquanto o nosso modelo é que podemos escrever um milhão de artigos e nós sabemos que praticamente todo mundo irá ler, porque cada uma daquelas histórias pode ser dimensionada para uma audiência tão pequena quanto uma pessoa. (MOSS, 2014)

Quanto mais dados estiverem a disposição dos algoritmos da *Automated Insights* <<http://automatedinsights.com/sports/>>, maior será a precisão e a qualidade do trabalho desempenhado pelo seu produto. No caso deste produto seu foco é processar grandes quantidades de dados e retornar para o seu público alvo uma informação precisa em linguagem natural²⁹, pesando o que é mais relevante para o seu tipo de leitor. De acordo com o diretor executivo, durante um jogo de futebol ao vivo, onde são captados dados das posições dos jogadores e pontuações em tempo real, é possível narrar o jogo em texto sem que sua audiência perceba uma grande diferença do que se fosse feito por um humano (MOSS, 2014). Supondo que isso já seja uma realidade e pudéssemos integrar a esse sistema um sintetizador de voz, teríamos um narrador nem um pouco carismático, mas com o mínimo de narrativa para construir um diálogo para um filme, por exemplo.

Novas possibilidades surgem neste momento. O **Body Labs** <<http://www.bodylabs.com/>> oferece um sistema através do kinect, capaz de captar os movimentos e o corpo de uma pessoa e reproduzi-los em um modelo 3D. Na possibilidade de compartilhar esse recurso na web abertamente, seria possível implementar mais poses para personagens ou modelos humanos para uso em filmes 3D, dinâmicos ou não. Vale lembrar que já existem dois formatos existentes que pensaram nessa possibilidade: O VRML (agora X3D) e COLLADA, ambos já estão alinhados com o padrão de marcação textual do HTML/XML para distribuição de dados na web, e podem ser compartilhados abertamente. Outra opção é por exemplo encapsular (integrar ao *Back-end*) softwares de animação abertos como o Blender em um serviço web e construir ou instalar um plugin capaz de exportar informações de animações nos mais variados formatos.

²⁹ Linguagem Natural - É uma sub-área da inteligência artificial e da linguística que estuda a compreensão e geração de línguas humanas naturais

4.4 - Futuro e Web 3.0

Até o momento a proposta comporta-se dentro do modelo da Web 2.0, onde o usuário está ativamente auxiliando na produção de novas informações, tornando essa base de dados cada vez mais rica. No futuro, a distribuição de animação e captura de movimentos pode seguir de um sistema ainda mais descentralizado, onde não só os usuários, mas os demais dispositivos conectados na internet sejam capazes de oferecer dados relevantes para a construção de um filme.

Smartphones, tablets, notebooks, drones, carros e toda a indumentária inteligente como relógios, roupas, calçados conectados à internet são sensores em potencial na coleta de dados que poderiam ser usados para poses de personagens, ângulos e estilos de câmera, vozes, posição geográfica. Muitas dessas informações já são captadas na indústria para a produção de filmes, e muitas vezes através de equipamentos caros e especializados. Com o avanço dos dispositivos a tendência é que essas técnicas se tornem mais acessíveis para as pessoas.

Cada um dos dispositivos inteligentes armazena os dados conforme as especificações de sua plataforma (sistema operacional) e software interno dedicado para suas funções. Todos esses dados acabam sendo inúteis se não há uma forma padronizada de os interpretar por toda a web. As APIs de dados que tornam possíveis os *mashups* buscam resolver esse problema. Independente do tipo de base de dados e plataforma que torne o YouTube possível, a sua API irá retornar na requisição os dados dentro de uma hierarquia lógica e semântica. Todos os outros serviços e dispositivos pela web serão capazes de fazer uso das informações.

O verdadeiro poder da Web Semântica será alcançado quando as pessoas criarem muitos programas que coletam conteúdo web de diversas fontes, processa a informação e troca os resultados com outros programas. A eficácia de tais agentes de software irá crescer exponencialmente conforme mais conteúdo Web legível por máquinas e serviços autômatos (incluindo outros agentes) se tornem disponíveis. A Web Semântica promove essa sinergia: mesmo agentes que não foram criados expressamente para para trabalhar em conjunto, podem transferir dados entre eles quando estão em formato semântico. (BERNERS-LEE, 2002)

O processo onde serviços na web compartilham dados em formatos padronizados e com significado multiplataforma é chamado de **Web Semântica**. Ao tornar dispositivos e serviços web interligados todos os dias, não só os usuários, mas as máquinas são capazes de aprender e se comunicar constantemente. Dessa forma, é possível utilizar a web fora do tradicional formato de se acessar a interface pela tela, que exige que o usuário esteja exercendo a tarefa específica de interagir com a aplicação.

Um exemplo de projeto que cria padrões para a integração de dados entre aplicações é o **CIDOC CRM** acessível no seguinte endereço na web: <<http://bit.ly/1LeKrxB>>, que é usado na área de gestão documental de museus. No CRM (página 22) é possível observar que essa forma em hierarquia de descrever objetos, busca encontrar um meio em que todo sistema de gestão de documentos seja capaz de “conversar” entre si ao compartilhar dados. O mesmo conceito, pode ser usado para descrever cenas, rigs³⁰, ações e seus quadro-chaves³¹ para a comunicação entre serviços web. Vários projetos de automação de produção de filmes poderiam trocar informações de suas bases de dados, para que cada uma fique melhor em suas próprias funções.

30 Ossos ou *Bones*, são utilizados para movimentar a malha de um personagem 3D.

31 Definem a pose de um personagem dentro de um espaço de tempo, geralmente é uma pose tomada como referência para o fluxo de uma animação.

5.0 - CONCLUSÃO

Nos últimos 25 anos, a web tem se tornado cada vez mais presente nas atividades sociais e econômicas. Aos poucos ela migrou de uma mera página na web para o corpo do então usuário, o que vem produzindo grandes quantidades de dados cuja utilidade chega a desafiar a imaginação. A complexa integração entre serviços web possibilitará avanços tecnológicos dentro da área de inteligência artificial. Dessa forma permitirá uma melhor interpretação das informações, e potencial utilidade ao cinema de animação. Utilidade para a elaboração mais ágil e individualizada de filmes para cada contexto, trazendo aos poucos uma parte do processo criativo para uma possível quarta revolução industrial (OBERHAUS, 2015).

A inteligência artificial das máquinas ainda não é complexa o suficiente para executar a linguagem cinematográfica tão bem como os humanos. Mas se espera que na próxima década a capacidade de aprendizado das máquinas passe a assumir funções consideradas intelectuais que exijam trabalho mental repetitivo. Atualmente apenas os humanos são capazes de criar ou resolver problemas com pensamento espontâneo.

Podemos considerar a Web 1.0, 2.0 e 3.0 como iterações que a cada volta trazem mais camadas de complexidade à comunicação na web, e consigo novas conexões entre humanos e máquinas através de aplicações e dispositivos inteligentes. As funções das máquinas começam a se tornar mais abstratas e descentralizadas, pedindo do artista no futuro maior sensibilidade em conhecimentos de informática para que suas próprias especialidades não se tornem alvos de automação.

6.0 - GLOSSÁRIO

C

CSV – *Comma-Separated Values* (valores separados por vírgulas): arquivo de texto para organização de dados, estruturado em quebras de linha e vírgulas.

COLLADA – *COLLABorative Design Activity* (atividade de design colaborativo): linguagem de marcação baseado em XML para descrever cenas e objetos tridimensionais.

M

MySQL – Base de dados relacional *open-source*, baseada em relacionamentos entre tabelas. Atualmente mantida pela Oracle. Site oficial: <<https://www.mysql.com>>

MongoDB – Base de dados de documento: armazena dados em listas de valores-chaves. Site oficial: <<https://www.mongodb.org/>>

N

Neo4j – Base de dados gráfica: armazena dados em formatos de vértices (*vertices*) conectados por arestas (*edges*). Site oficial: <<http://neo4j.com/>>

O

OrientDB – Base de dados gráfica e documento: armazena dados em formatos de vértices conectados por arestas, que também abrigam listas de chaves-valores. Site oficial: <<http://orientdb.com/>>

P

PHP – *Hypertext Preprocessor*. Linguagem de programação *Back-end* orientada a objetos, desenvolvida em 1994 para servir páginas dinâmicas. Site oficial: <<http://php.net>>

Python – Linguagem de programação *Back-end* orientada a objetos de propósito geral, também implementada na web. Site oficial <<https://www.python.org/>>

Postgres – Base de dados relacional *open-source*. Site oficial: <<http://www.postgresql.org/>>.

R

Ruby – Linguagem de programação *Back-end* orientada a objetos. Utiliza o *framework* Rails para servir páginas dinâmicas para a web. Site oficial: <<https://www.ruby-lang.org/pt/>>

X

X3D – Sucessor do VRML.

XHTML – eXtensible Hypertext Markup Language, versão do HTML que incorpora elementos do XML.

XML – *eXtensible Markup Language*: linguagem de marcação utilizada para organização de dados.

V

VRML – *Virtual Reality Markup Language* (Linguagem de Marcação de Realidade Virtual). Linguagem de marcação usada para descrever objetos tridimensionais para uso em realidade virtual. Antecessora do X3D.

7.0 - BIBLIOGRAFIA

LIALINA, Olia. *Digital Folklore*, 2009, visualizado em 07/07/2015. Disponível em: <<http://digitalfolklore.org/>>

MANOVICH, Lev. *The language of New Media*. Cambridge, MIT Press, 2001.

MANOVICH, Lev. *Database as a genre of new media*, 1997, visualizado em 07/07/2015. Disponível em <http://time.arts.ucla.edu/AI_Society/manovich.html>

O'REILLY, Tim. *What is Web 2.0*, 2005, visualizado em 07/07/2015. Disponível em <<http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>>

MEI, Fabrizio. *O que é Render?* visualizado em 07/07/2015. Disponível em: <<http://chocoladesign.com/o-que-e-render>>

CAMARGO, Camila. *O que é cliente-servidor?* visualizado em 07/07/2015.. Disponível em: <<http://www.tecmundo.com.br/982-o-que-e-cliente-servidor-.htm>>

BERNERS-LEE, Tim. *The Semantic Web*. Scientific American, 2002.

REILLY, Claire. *Absolutely fabulist: The computer program that Writes fables*, 2015, visualizado em 07/07/2015. Disponível em: <<http://www.cnet.com/news/absolutely-fabulist-computer-program-writes-fables/>>

MOSS, Richard. *Creative AI: Teaching computers to be reporters and storytellers*, visualizado em 07/07/2015.. Disponível em: <http://www.gizmag.com/creative-ai-automated-writing-storytelling/35989/>

COUCHOT, Edmond. *A segunda interatividade* In: DOMINGUES, Diana. *Arte e vida no século XXI*, p. 27, 2003.

WESLEY, Addison. *A history of HTML*, 1998, visualizado em 07/07/2015. Disponível em: <http://www.w3.org/People/Raggett/book4/ch02.html>

LOMAS, Natasha. *Amazon patents anticipatory shipping*, 2014, visualizado em 07/07/2015. Disponível em: <<http://techcrunch.com/2014/01/18/amazon-pre-ships/>>

BOS, Bert. *Cascading Style Sheets, designing for the Web*, 1999, visualizado em 07/07/2015. Disponível em: <<http://www.w3.org/Style/LieBos2e/history/>>

Richard. *Plugins vs Extensions – the difference*, 2010, visualizado em 07/07/2015. Disponível em: <<http://colonelpanic.net/2010/08/browser-plugins-vs-extensions-the-difference/>>

CHAMPEON, Steve. *Javascript: How did we get here?* 2001, visualizado em 07/07/2015. Disponível em: <http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html>

THOMSON, Rebecca. *How technology is improving air traffic control*, 2010, visualizado em 07/07/2015. Disponível em: <<http://www.computerweekly.com/feature/How-technology-is-improving-air-traffic-control>>

W3C, *A vocabulary and associated APIs for HTML and XHTML*, 2014, visualizado em 07/07/2015. Disponível em: <<http://www.w3.org/TR/html5/introduction.html>>

LEONHARD, Gerd. *Big Data, big business, big brother?* 2014, visualizado em 07/07/2015. Disponível em: <<http://edition.cnn.com/2014/02/26/business/big-data-big-business/index.html>>

OBERHAUS, Daniel. *This is what the fourth industrial revolution looks like*, visualizado em 07/07/2015.. Disponível em: <<http://motherboard.vice.com/read/life-after-the-fourth-industrial-revolution>>

THE JARGON FILE. *September that never ended*, visualizado em 01/04/2015. Disponível em: <<http://www.catb.org/jargon/html/S/September-that-never-ended.html>>