

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

LUCIAN ROSSONI RIBAS

**IMAGEJS: FERRAMENTA JAVASCRIPT *CROSS-
BROWSER* PARA COMPOSIÇÃO DE UM MINI
ESTÚDIO PARA EDIÇÃO DE IMAGENS**

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2015

LUCIAN ROSSONI RIBAS

**IMAGEJS: FERRAMENTA JAVASCRIPT *CROSS-
BROWSER* PARA COMPOSIÇÃO DE UM MINI
ESTÚDIO PARA EDIÇÃO DE IMAGENS**

Projeto de Trabalho de Conclusão de Curso, apresentado à disciplina de Trabalho de Conclusão de Curso, do Curso Superior de Tecnologia em Sistemas para Internet – da Universidade Tecnológica Federal do Paraná como requisito parcial para aprovação.

Orientadora: Profa. Dra. Simone Bello Kaminski
Aires

Coorientador: Prof. Me. Emerson Andre Fedechen

GUARAPUAVA

2015

RESUMO

RIBAS, Lucian. IMAGEJS: Ferramenta JavaScript Cross-Browser para Composição de um Mini Estúdio para Edição de Imagens. 36 f. Projeto de Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná – UTFPR. Guarapuava, 2015.

Atualmente, os principais recursos desenvolvidos para edição de imagens são direcionados para dispositivos *desktops*. Porém, o aumento da capacidade computacional aplicada em recursos para Internet está desencadeando lentamente a migração dos programas *desktops* para programas *web*. Isto inclui os *softwares* desenvolvidos para manipulação de imagens, definindo o escopo deste projeto. Disponibilizado no formato de *framework*, com bibliotecas e dependências *javascript*, este recurso tem como objetivo incorporar a uma página HTML, um pequeno estúdio para manipulação de imagens. Agrupando, de forma gráfica, várias ferramentas para aplicar edições básicas em uma imagem. A imagem poderá ser selecionada pela interface, por uma opção própria, ou enviada a partir de um endereço *web* externo, onde o usuário terá como retorno a imagem com as edições aplicadas. Por se tratar de uma biblioteca *cross-browser*, o recurso poderá auxiliar desenvolvedores de sistemas para Internet, sendo que estes não precisarão implementar módulos referentes a edição básicas de imagens, desde que disponham do *framework* oferecido por este projeto.

Palavras-Chave: Desenvolvimento *Web*, *Front-End*, *Framework*, *Cross-Browser*.

ABSTRACT

RIBAS, Lucian. IMAGE JS: Framework JavaScript Cross-Browser Composition for a Mini Studio Image Editing. 36 f. Working project of Course - Course of Technology in Internet Systems, Federal Technological University of Paraná - UTFPR. Guarapuava, 2015.

Currently, the key features developed for image editing are targeted to desktop devices. However, the increase in computing power applied to Internet resources is slowly triggering the migration of desktop programs for web programs. This includes software developed for image manipulation, defining the scope of this project. Available in the framework format, with javascript libraries and dependencies, this feature aims to incorporate into an HTML page, a small studio for image manipulation. Grouping, graphically, various tools to apply basic edits on an image. The image may be selected by the interface, by its own choice, or sent from an external web address where it will have to return the picture with edits applied. Since this is a cross-browser library, the resource will be useful for Internet systems developers, and these will not need to implement basic issue concerning modules images, provided they have the framework offered by this project.

Keywords: Web Development, Front End, Framework, Cross-Browser.

LISTA DE SIGLAS

API	Interfaces de programação de aplicativos.
CSS	Folhas de estilo em cascata (do inglês <i>Cascating Style Sheets</i>)
DOM	Document Object Model (Em português: Modelo de Objeto de Documentos).
HTML	Linguagem de Marcação de Hipertexto (do inglês <i>HyperText Markup Language</i>).
IHC	Interação humano-computador (também conhecida como interação homem-computador) é o estudo da interação entre pessoas e computadores.
JS	Abreviação para a linguagem JavaScript.
TCC	Trabalho de Conclusão de Curso.
URL	Localizador Padrão de Recursos (do inglês <i>Uniform Resource Locator</i>) é o endereço de um recurso disponível em uma rede.

LISTA DE FIGURAS

Figura 1 – Exemplo de Estúdio Editor de Imagens (befunky).....	10
Figura 2 – Ferramenta StudioJS.....	12
Figura 3 – Ferramenta CamanJS.....	13
Figura 4 – Ferramenta PIXLR.....	14
Figura 5 – Exemplo de Interface Construída com Material Design	18
Figura 6 – Protótipo de tela do ImageJS.....	25
Figura 7 – ImageJS Incorporado, esquema de interações.....	28
Figura 8 – ImageJS Online, esquema de interações.....	31
Figura 9 – Protótipo de página inicial do ImageJS online.....	32
Figura 10 – Protótipo de página inicial do ImageJS online.....	32

SUMÁRIO

1 INTRODUÇÃO.....	7
1.1 OBJETIVOS.....	9
1.1.1 Objetivo Geral.....	9
1.1.2 Objetivos Específicos.....	9
1.2 DIFERENCIAL TECNOLÓGICO.....	9
2 RESENHA LITERÁRIA.....	11
2.1 ESTADO DA ARTE.....	11
2.1.1 A FERRAMENTA STUDIOJS.....	11
2.1.2 A BIBLIOTECA CAMANJS.....	12
2.1.3 O EDITOR PIXLR.....	13
2.2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.2.1 HYPERTEXT MARKUP LANGUAGE (HTML).....	15
2.2.2 CASCADING STYLE SHEET (CSS).....	16
2.2.3 FRAMEWORK FRONT-END MATERIALIZE.....	17
2.2.4 LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT.....	19
2.2.5 A BIBLIOTECA JQUERY.....	20
3 METODOLOGIA.....	22
3.1 LEVANTAMENTO DE REQUISITOS.....	22
3.2 LISTAGEM DAS FERRAMENTAS.....	22
3.2.1 FERRAMENTAS DE EDIÇÃO.....	24
3.2.2 FERRAMENTAS TEXTUAIS.....	25
3.2.3 FILTROS.....	25
3.3 PROJETAR E DESENVOLVER A INTERFACE.....	26
3.4 PROJETAR E DESENVOLVER FERRAMENTAS.....	27
4 DESENVOLVIMENTO.....	28
4.1 EDITOR INCORPORADO.....	28
4.2 EDITOR ONLINE.....	30
4.3 ARQUITETURA DO SISTEMA.....	32
5 CRONOGRAMA.....	33
6 CONSIDERAÇÕES FINAIS.....	34
REFERÊNCIAS.....	35

1 INTRODUÇÃO

Atualmente é comumente pessoas com um computador em mãos poderem elaborar imagens, vídeos, fotografias, filmes, entre outros dados no formato de mídias digitais. A necessidade da manipulação gráfica destes dados tornou-se responsabilidade da computação presente no dia-a-dia. Ou seja, recursos simples, de fácil acesso, e principalmente, que atendam as necessidades mínimas do usuário rotineiro.

Devido a esta necessidade, existem diversos recursos computacionais desenvolvidos para prover as soluções para estes tipos de problemas. Muitos *softwares*, são desenvolvidos e disponibilizados para praticamente todos os ambientes de usabilidade, tais como, dispositivos móveis, computadores *desktop*, entre outras classes aparelhos.

A Internet certamente é o meio com maior abundância de aplicativos, devido a sua abrangência. A quantidade de usuários que acessam estes programas está aumentando em simultâneo com os recursos disponibilizados em *nuvem*, assim como afirma a matéria publicada por Marcel Gugoni, no site AMCHAM BRASIL¹.

O usuário rotineiro está se adaptando ao uso de sistemas disponíveis na Internet para auxiliar as atividades e interações virtuais, fazendo com que estes tipos de recursos se tornem comuns no seu dia-a-dia. Por exemplo, segundo a matéria publicada pela página *Terra*², em 2013, o Google (sistema para pesquisas *online*) recebe por dia aproximadamente 3 bilhões de pesquisas, onde 15% destas são inéditas.

Percebe-se que muitos dos sistemas presentes na *web*, possuem alguma relação com arquivos no formato de mídias³, seja para o envio de desenhos, imagens ou fotos. Devido a isto, seria interessante que estes sistemas possuam alguma etapa para efetuar pré edições nas imagens que forem enviadas pelos usuários. Porém, atualmente os sistemas *web* que possuem

1 <http://www.amcham.com.br/business-in-growth/noticias/migrar-para-a-nuvem-ajuda-a-economizar-ate-90-nos-gastos-de-ti>

2 <http://idgnow.com.br/internet/2013/09/25/busca-google-15-das-pesquisas-diarias-nunca-foram-feitas-antes>

3 <http://oglobo.globo.com/sociedade/tecnologia/por-ano-125-bilhoes-de-imagens-sao-compartilhadas-na-rede-8301345>

algum recurso, tratam apenas de edições extremamente básicas como: recortes, filtros, controle de brilho e contrastes. Ainda, muitos dos sistemas, oferecem apenas a opção de envio de imagens, considerando que as pré edições já tenham sido feitas pelo próprio usuário.

Em virtude da dificuldade de desenvolvimento ou até mesmo por questões financeiras, há poucos recursos disponíveis neste contexto para a utilização por programadores de sistemas *web*. A muitas ferramentas disponíveis atendem apenas aos princípios básicos de edição. Logo, a maioria dos sistemas não possuem este recurso, pois para desenvolver esta ferramenta do início da implementação seria necessário muito mais tempo, conhecimento e mão de obra aplicada no projeto.

Para poupar tempo de desenvolvimento, utiliza-se *frameworks* que têm como objetivo apoiar o desenvolvedor na construção do projeto, oferecendo uma série de códigos e funcionalidades prontas e testadas. Segundo Fayad e Schmidt (1997), *frameworks* são um conjunto de classes pré-fabricadas de *software*, que trabalham em conjunto para realizar uma ou mais responsabilidades para um domínio de um subsistema da aplicação.

O escopo deste trabalho é o desenvolvimento do *framework* ImageJS, que será uma ferramenta para utilização através de *browsers* (navegadores). Não é possível identificar qual navegador está acessando a biblioteca em cada requisição, logo, o *framework* deve ser genérico neste contexto, ou seja, uma ferramenta *cross-browser*. Segundo a matéria publicada por Miguel Angel, no site Criar Web⁴, soluções *cross-browser* se adaptam a todo tipo de *browsers* (navegadores).

1.1 OBJETIVOS

Nesta seção serão apresentados o objetivo geral e objetivos específicos do presente trabalho.

4 <http://www.criarweb.com/artigos/o-que-e-cross-browser.html>

1.1.1 Objetivo Geral

O objetivo geral deste projeto consiste em desenvolver um *framework* JavaScript incorporável (*cross-browser*), compondo em uma página *web* um mini estúdio para aplicar edições básicas em uma imagem.

1.1.2 Objetivos Específicos

Apresenta-se os seguintes itens como objetivos específicos desde trabalho:

- Projetar e desenvolver um *framework* com para edição de imagens;
- Estudar técnicas de processamento de imagem;
- Projetar e desenvolver um mini estúdio;
- Desenvolver um site realizando a simulação do uso do *framework* para validação do trabalho;

1.2 DIFERENCIAL TECNOLÓGICO

Um dos principais diferenciais deste projeto, é a possibilidade de trabalhar de forma adaptável para qualquer formulário da *web*. Onde, por exemplo, a única configuração necessária para que seja possível o *framework* selecionar a imagem que será enviada para edição é a identificação do elemento HTML referente ao *upload* da imagem (*input* do tipo *file*). No momento do envio, o *framework* irá automaticamente abrir a imagem para que o usuário efetue as edições.

O recurso está disponível ao usuário no formato de estúdio para edição de imagens, e não apenas como bibliotecas implementáveis. Para melhor compreensão do que se trata um estúdio de imagens foi destacado um exemplo na Figura 1.

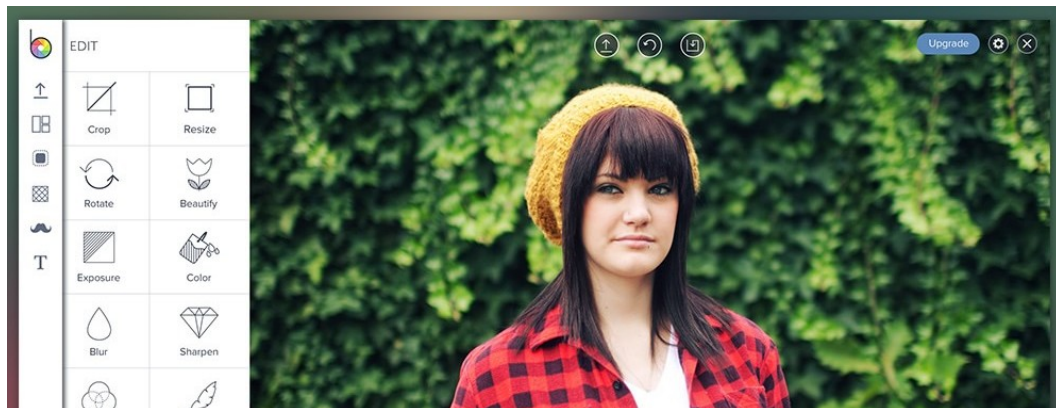


Figura 1 – Exemplo de Estúdio Editor de Imagens (Befunky)

Fonte: Befunky, Página Inicial

- Outro diferencial presente no projeto, trata-se da sua interface leve e objetiva baseada nos estilos Material⁵, *framework front-end*⁶ desenvolvido pela Google.
- Uma vantagem interessante desta ferramenta, se trata do seu código ser aberto ou *open-source*, possibilitando que a comunidade interessada possa contribuir no desenvolvimento do projeto.

5 <https://design.google.com/>

6 *Front-end*: Parte visual de um sistema, área de entrada de dados dos usuários.

2 RESENHA LITERÁRIA

Neste Capítulo serão abordados os trabalhos já desenvolvidos na área do projeto. Assim, como as tecnologias que serão utilizadas para o desenvolvimento do *framework*.

2.1 ESTADO DA ARTE

Nesta seção serão abordados trabalhos correlatos, sendo apresentado os mais relevantes a essa pesquisa e desenvolvimento.

2.1.1 A FERRAMENTA STUDIOJS

O primeiro trabalho de referência, se chama *StudioJS*, trata-se de um projeto de desenvolvimento *cross-browser* com código aberto e livre, no mesmo ramo e aplicação do trabalho proposto. Porém não possui ferramentas textuais como inserção e definição de estilos textuais (negrito, itálico, etc). Está ferramenta também não utiliza nenhum *framework* popular para estilização da interface e segundo testes aplicados pelo autor apresenta lentidão ao aplicar suas funcionalidades.

Disponibilizado pela empresa IMAZEM⁷, gerenciadora de *softwares* para processamento de imagem. Uma das principais vantagens da ferramenta, é o fato dela estar apresentada no formato de um pequeno estúdio de imagens, possibilitando maior usabilidade ao usuário final. Porém, sua interface não é muito interativa, oferecendo poucas opções ao usuário. A Figura 2, apresenta uma tela da ferramenta StudioJS.

7 <http://studio.imageresizing.net/studio.html>

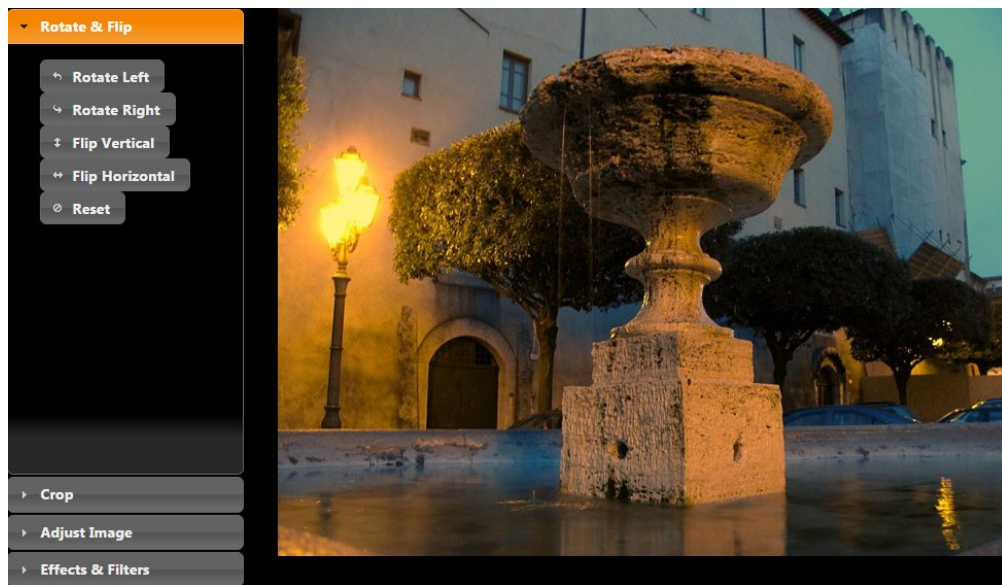


Figura 2: Ferramenta StudioJS

Fonte: Imazen, StudioJS.

2.1.2 A BIBLIOTECA CAMANJS

A ferramenta *CamanJS* foi desenvolvida por Ryan LeFevre⁸. Este projeto conta com várias contribuições de outros programadores, assim como o trabalho de referência anterior (*StudioJS*). Este recurso não está apresentado no formato de estúdio, mas sim como bibliotecas independentes. Ela oferece várias ferramentas de extrema eficiência perante edições básicas de imagens. Caso o desenvolvedor deseje compor um estúdio com estas ferramentas, deverá implementar sua própria solução.

Esta biblioteca foi implementada utilizando o recurso Canvas⁹ do HTML 5, a mesma tecnologia que está proposta como solução para o projeto ImageJS, porém o ImageJS possuirá uma gama maior de ferramentas , pois terá ferramentas textuais, de recortes, entre outros.

A Figura 3 representa uma tela com algumas das opções da biblioteca CamanJS.

8 <http://camanjs.com/>

9 Canvas: Elemento HTML utilizado para “desenhar“ usando uma linguagem *script*, normalmente o JavaScript.



Figura 3: Ferramenta CamanJS

Fonte: CamanJS, Página Inicial.

2.1.3 O EDITOR PIXLR

A ferramenta chama-se PIXLR, é uma ferramenta *web* não *cross-browser*, ou seja, é uma ferramenta para edição de imagens *online*, onde o usuário envia suas imagens, as manipula, e recupera a imagem com as edições efetuadas. Desenvolvida pela empresa Autodesk¹⁰, o recurso PIXLR está disponível para utilização via aplicativo *mobile*, programa *desktop* ou sistema *web*.

Sendo o seu *layout*¹¹ a principal referência para este projeto, o PIXLR possui uma interface de fácil compreensão e usabilidade pelo usuário final. Pelo fato de ser uma ferramenta não *cross-browser*, sua biblioteca de opções é muito rica disponibilizando várias opções para efetuar edição e manipulação nas imagens enviadas pelo usuário. Na Figura 4, é possível observar a tela de edição da ferramenta PIXLR.

¹⁰ <http://www.autodesk.com.br/>

¹¹ *Layout*: Desenho, modelo da página Web.

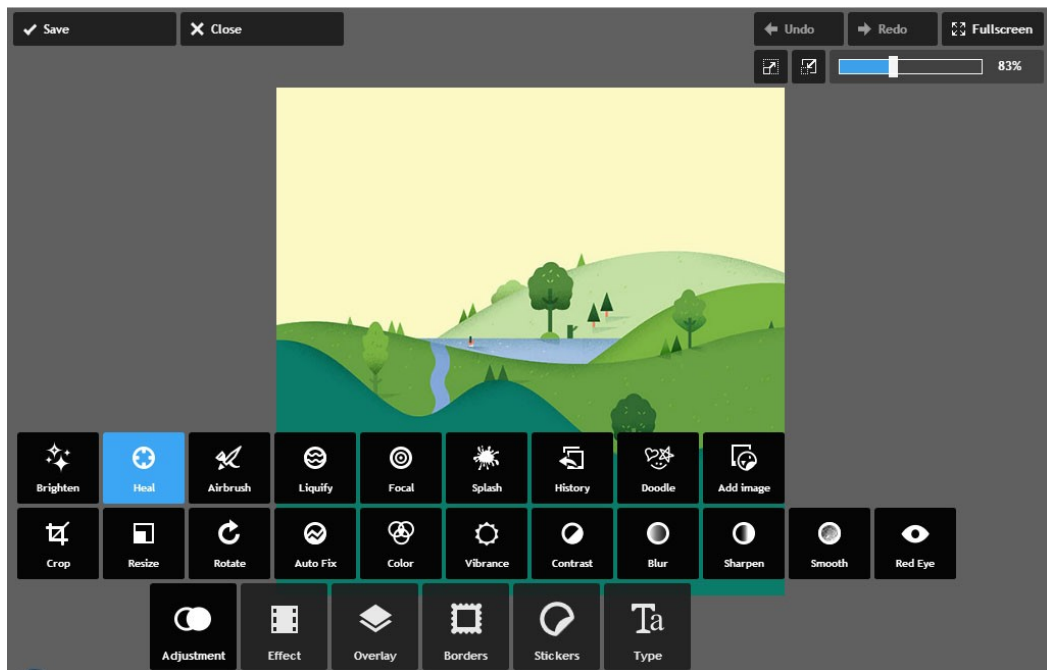


Figura 4: Ferramenta PIXLR

Fonte: PIXLR, PIXLR Express.

Os sistemas (PIXLR) e a biblioteca (StudioJS, CamanJS) citados acima, cumprem com seu propósito, que são oferecer códigos pré-programados e testados, implementando funcionalidades de aplicabilidade à edições básicas de imagens. E por essa razão o trabalho proposto ImageJS, terá como base de desenvolvimento algumas funcionalidades disponíveis nestas ferramentas. A Tabela 1 representa um comparativo com as funcionalidades disponíveis entre estas ferramentas e o ImageJS.

Tabela 1: Comparativo de funcionalidades entre os sistemas

Funcionalidade	StudioJS	CamanJS	PIXLR	ImageJS
Seletor de área retangular	Não possui	Não possui	Possui	Possuirá
Recorte retangular	Possui	Não possui	Possui	Possuirá
Controle de brilho	Possui	Não possui	Possui	Possuirá
Controle de Contraste	Possui	Não possui	Possui	Possuirá
Rotação para esquerda	Possui	Não possui	Possui	Possuirá
Rotação para direita	Possui	Não possui	Possui	Possuirá
Redimensionador retangular	Não possui	Não possui	Possui	Possuirá

Inserção de Textos	Não possui	Não possui	Possui	Possuirá
Controle do tamanho textual	Não possui	Não possui	Possui	Possuirá
Transformação textual em negrito	Não possui	Não possui	Possui	Possuirá
Transformação textual em itálico	Não possui	Não possui	Possui	Possuirá
Sublinhamento textual	Não possui	Não possui	Possui	Possuirá
Alinhamento textual	Não possui	Não possui	Possui	Possuirá
Controle de cores textuais	Não possui	Não possui	Possui	Possuirá
Filtros e Efeitos	Possui	Possui	Possui	Possuirá
Interface responsiva	Não possui	Não possui	Possui	Possuirá
<i>Cross-browser</i>	Possui	Possui	Não Possui	Possuirá
<i>Framework</i> de funcionalidades	Não possui	Possui	Não possui	Possuirá

2.2 FUNDAMENTAÇÃO TEÓRICA

Neste item são definidos os principais conceitos que estão relacionados com este trabalho e as tecnologias que serão utilizadas em seu desenvolvimento.

2.2.1 HYPERTEXT MARKUP LANGUAGE (HTML)

Uma página inicial, conhecida como *home page* ou página web, é um arquivo textual comum e armazenado na extensão .HTM ou .HTML cujo o conteúdo é composto basicamente de textos e códigos especiais chamados tags que possibilitam a exibição destes arquivos na web (World Wide Web) através de um programa especial chamado navegador ou Agente de Usuário¹². É possível encontrar páginas na Internet com outras extensões, como .ASP, .JSP entre outras, criadas por outras tecnologias. Além de texto, este tipo de documento pode conter referências para imagens, som, animações, vídeos e até mesmo *links* ou ligações com

¹² Agente de usuário: Termo empregado para se fazer referência a qualquer dispositivo capaz de interpretar um documento escrito em linguagem de marcação. O exemplo mais comum e conhecido de agente de usuário é o navegador. Leitores de tela, robôs de indexação e busca, dispositivos móveis como tablets e *smarthphones* são também alguns exemplos de agentes de usuário (SILVA, 2012).

outras páginas armazenadas em outros endereços web (RAMALHO, 2005).

Para compartilhar informações de maneira universal, é necessário haver um padrão que seja entendido por diversos meios de acesso. O HTML se propõe a ser a linguagem com este padrão. Desenvolvido originalmente por Tim Berners-Lee o HTML ganhou popularidade quando o Mosaic – browser desenvolvido por Marc Andressen na década de 1990 – ganhou força (W3C, 2010).

O HTML ganhou as versões HTML+, HTML 2.0 e HTML 3.0 entre 1993 e 1995, onde foram propostas várias mudanças para o enriquecimento da linguagem. Contudo, até então, o HTML não era tratado como um padrão no mundo da web. Apenas em 1997, o grupo de trabalho da W3C responsável por manter o padrão do código, lançou a versão 3.2 da linguagem, fazendo com que ela se torna uma prática comum na *web* (W3C, 2010).

O Principal objetivo do HTML 5 é facilitar a manipulação de elementos e objetos do DOM (Modelo de Objeto de Documentos, do inglês *Document Object Model*). Ao contrário das versões antecessoras, o HTML 5 disponibiliza ferramentas para que a CSS (Folhas de estilo em cascata, do inglês *Cascading Style Sheets*) e o JavaScript (seção 2.2.4) da página cumpram com seu trabalho da melhor maneira possível. O HTML permite por meio das suas APIs (Interfaces de Programação de Aplicativos, do inglês *Application Program Interface*) a manipulação das características destes elementos, de forma que a página web continue leve e funcional. As versões antigas do HTML não continham um padrão universal para a criação de seções comuns e específicas como rodapé, cabeçalho, sidebar, menus e etc. Não havia um padrão de nomenclatura de ID, Classe (ID e Classe são atributos de um elemento HTML) ou tags. Não havia um método de capturar de maneira automática as informações localizadas nos rodapés dos websites. O HTML5 propôs uma forma mais organizada e semântica de efetuar a marcação nos textos (W3C, 2010).

Neste projeto o papel do HTML no lado do cliente será marcar o elemento de entrada da imagem, um *input* de dados, do tipo *file* (arquivo). No escopo interno do *framework*, o objetivo do HTML será estruturar os elementos do estúdio, tais como seus botões, menus, áreas de edições, entre outros elementos. [METODOLOGIA]

2.2.2 CASCADING STYLE SHEET (CSS)

Traduzido do português como folhas de estilo em cascata. Tem por finalidade aplicar à marcação HTML/XML o estilo visual dos elementos. Por definição, não cabe ao HTML fornecer informações ao Agente do Usuário sobre a apresentação dos elementos, como por exemplo: cores de fontes, tamanhos de textos, posicionamento dos demais aspectos visuais de um documento. Cabem à CSS todas as funções de apresentação de um documento (SILVA, 2012).

Setembro de 1994, surge a primeira proposta de implementação da CSS. Mas apenas em 17 dezembro de 1996 a CSS foram lançadas como uma recomendação oficial do W3C, padronizando a sua primeira versão, as CSS1 (SILVA, 2012).

Atualmente, a CSS se encontram na sua versão 3, onde sua principal diferença é o modelo de desenvolvimento adotado. Enquanto as versões anteriores da CSS adotaram modelos de desenvolvimento baseados em um único documento, as CSS3 estão sendo desenvolvidas em módulos, permitindo o desenvolvimento independente de cada módulo, possibilitando que os fabricantes implementem funcionalidades previstas em módulos futuros, adiantando o desenvolvimento do Agente de Usuário em questão (SILVA, 2012).

Neste projeto a CSS será responsável por estilizar todo o conteúdo do ImageJS, desde cores, bordas, aspecto dos botões, menus, posicionamento e comportamento de alguns elementos HTML, entre outros aspectos visuais. A maneira que será estruturado a CSS do projeto definirá o conforto visual e usabilidade que o usuário terá sobre o estúdio, proporcionando uma experiência satisfatória no seu uso. **[METODOLOGIA]**

O desenvolvimento das aplicações que usam CSS pode ser facilitado por ferramentas e *frameworks front-end*. Neste projeto, devido a sua interface, simples de usar e com estilos baseados nas interfaces Material Design, adota-se o Materialize como *framework* padrão da interface.

2.2.3 FRAMEWORK FRONT-END MATERIALIZE

Segundo a Google¹³, empresa desenvolvedora dos estilos visuais Material Design¹⁴, a idéia do projeto Material é proporcionar aos seus usuários uma experiência visual inovadora, com um padrão de linguagem de apresentação que se aplique com os princípios clássicos de um bom *design*, oferecendo inovação, tecnologia e possibilidade de aplicação de ciência (aplicação de cores e formatos mais aceitos pelo olho humano) nos aspectos dos elementos.

Devido à boa aceitação pelos usuários do modelo visual Material, a comunidade web passou a desenvolver vários *frameworks front-end* que utilizam dos conceitos e modelos visuais do Material, como o próprio Materialize¹⁵ que será utilizado neste projeto.

O Materialize possui em sua composição um agrupamento de várias bibliotecas CSS e JavaScript que foram desenvolvidas pela comunidade, tornando-se bastante modular e possibilitando que o usuário controle os recursos que são inclusos na sua aplicação, requerendo todos os *scripts* ou apenas os componentes das funções que deseja utilizar. No site¹⁶ oficial do *framework* é possível visualizar corretamente toda sua arquitetura.

Um *framework* raramente consegue atender a todas as necessidades dos usuários. Por tanto, alguns elementos serão editados ou criados do início de acordo o desenvolvimento e necessidade do projeto. Como por exemplo, para os elementos *dropdowns* (menus) foram criados estilos CSS e comportamentos JavaScript do início. O objetivo do Materialize não é propor toda a interface visual e comportamental, mas apenas fornecer a estrutura básica de apresentação do ImageJS. Na Figura 5, é apresentada uma tela construída com os estilos do Material Design.

13 <https://www.google.com.br/>

14 <https://www.google.com/design/spec/material-design/introduction.html>

15 <http://materializecss.com/>

16 <http://materializecss.com/getting-started.html>

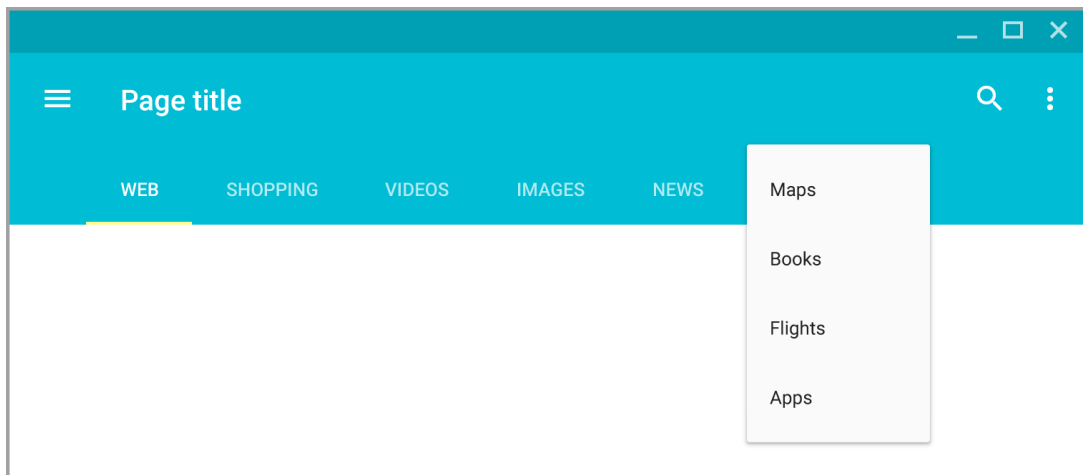


Figura 5 – Exemplo de Interface Construída com Material Design

Fonte: Google, Material Design.

A Figura 5, representa uma tela protótipo, ou seja, desenvolvida apenas para representação visual dos estilos Material Design. A tela em si não possui relação com a ferramenta ImageJS.

2.2.4 LINGUAGEM DE PROGRAMAÇÃO JAVASCRIPT

A linguagem JavaScript foi criada pela Netscape, para controle de objetos XMLHttpRequest¹⁷ do DOM. Ao início do projeto chamava-se LiveScript, uma linguagem baseada no C, no C++ e no Perl. Dado que no mesmo período foi introduzido a linguagem Java (REMOALDO, 2008).

Segundo Crane et al. (2008), o JavaScript é uma linguagem de uso global de descendência híbrida, com uma pequena semelhança com a família das linguagens C. A JavaScript, pode ser considerado basicamente como uma linguagem de criação de *scripts* de uso geral, interpretada e fracamente tipada. Fracamente tipada significa que as variáveis não são declaradas especificamente como *strings*, inteiros ou objetos e que é possível atribuir valores de diferentes tipos à mesma variável. No Quadro 1, observa-se um código válido.

¹⁷ XMLHttpRequest: Objeto JavaScript responsável pela comunicação assíncrona com um servidor web.

A JavaScript chamava-se LiveScript, baseada nas linguagens C, C++ e Perl. Dado que no mesmo tempo foi introduzida a linguagem Java, e através de um acordo com a Sun, a Netscape¹⁸ resolveu mudar o nome da linguagem para JavaScript e disponibilizou a versão 1.0 em março de 1996 no Navigator 2.0 (REMOALDO, 2008).

Devido ao fato da Netscape não ter licenciado o JavaScript, em 1996 a Microsoft¹⁹ lançou uma linguagem chamada JScript para atribuir ao Internet Explorer 3.0. As linguagens tinham suas semelhanças, mas seus *scripts* não eram iguais, gerando problemas de incompatibilidades nas páginas web, obrigando os desenvolvedores a implementar dois códigos diferentes quando seus usuários utilizam diferentes navegadores (REMOALDO, 2008).

Em 1997 foi lançada uma norma chamada ECMA-262 pela ECMA²⁰, que tem como objetivo incorporar os elementos tanto do JavaScript como do Jscript em uma linguagem chamada ECMAScript (REMOALDO, 2008).

Neste projeto a linguagem JavaScript terá como responsabilidade toda a área dinâmica do *framework* ImageJS desde a abertura da imagem no editor até a aplicação de filtros e das ferramentas de edição.

Assim como em outros escopos, o desenvolvimento das aplicações que usam JavaScript pode ser facilitado por ferramentas e *frameworks front-end* criados para auxiliar o programador. Neste projeto, devido a variedade de códigos e funcionalidades implementadas pela comunidade será utilizado o *framework* JavaScript conhecido como jQuery, que é uma biblioteca *Facade*²¹ do JavaScript.

2.2.5 A BIBLIOTECA JQUERY

A jQuery é uma biblioteca JavaScript desenvolvida para auxiliar web designers e desenvolvedores web a criar e ampliar as interações JavaScript de maneira rápida e concisa, usando um conjunto definido de métodos que envolvem as funções nativas do JavaScript. A biblioteca jQuery não oferece qualquer funcionalidade nova, ela usa as APIs existentes do

18 <http://isp.netscape.com/>

19 <https://www.microsoft.com/>

20 ECMA: Associação Europeia de Fabricantes de Computadores (do inglês Europe Computer Manufacturers Association) – uma associação internacional de indústrias sem fins lucrativos.

21 Facade: Padrão de projeto que propõe uma interface otimizadora de funcionalidade de uma API.

JavaScript, que são cansativas e repetitivas, e as disponibiliza para um público mais amplo por meio de sua sintaxe de fácil compreensão e criação (RUTTER, 2012).

Devido à sua flexibilidade e sintaxe amigável, o jQuery acabou se tornando a biblioteca JavaScript mais utilizada pelos sistemas web. Em 2008 passou a ser a biblioteca padrão do ASP.NET MVC, da Microsoft, e o Ruby on Rails, a partir da versão 3, substituiu o Prototype pelo jQuery (BALDUINO, 2012).

Atualmente o jQuery se encontra na versão 2 e sua parte fundamental ainda são os *selectors*. Os *selectors* indicam o quê será utilizado. Basicamente, se entender o conceito de *selectors*, entendeu a maior parte do funcionamento do *framework* (BALDUINO, 2012).

O jQuery possui uma biblioteca implementada pra manipulação de imagem chamada jCanva²². Segundo seu site oficial se trata de uma biblioteca JavaScript, escrita usando jQuery e para jQuery, que envolve a API canvas HTML5, adicionando novos recursos e capacidades, muitos dos quais são customizáveis. Os recursos incluem camadas, eventos, drag-and-drop, animação, entre outros.

3 METODOLOGIA

Este Capítulo apresenta os passos a serem executados a fim de atingir os objetivos desse projeto.

3.1 LEVANTAMENTO DE REQUISITOS

Nesta etapa deverá ser definido os requisitos do sistema, ou seja todas as atividades e comportamentos que a ferramenta deverá possuir. Para isto será efetuado uma pesquisa baseando-se em outros *softwares* das mesma área (assim como os citados no Estado da Arte, seção 2.1) verificando quais são as ferramentas e os comportamentos mais presentes nestas programas. Com isto, espera-se ter embasamento suficiente das possibilidades de implementação nos editores de imagens, para assim poder desenvolver uma ferramenta de

²² <http://projects.calebevans.me/jcanvas/>

maior abrangência e genericidade.

3.2 LISTAGEM DAS FERRAMENTAS

As ferramentas para edição de imagens são amplamente utilizadas para preparar imagens antes de se postar em sistemas que são utilizados por vários usuário. Existem várias ferramentas disponíveis para efetuar vários tipos edições em imagens, desta maneira, para efetuar uma listagem das ferramentas mais relevantes para implementação, foram analisadas alguns programas que já possuem popularidade perante os usuários, que são: Photoshop²³, Photoscape²⁴, Instagram²⁵, Befunky²⁶ e PIXLR²⁷.

Na Tabela 2, são apresentadas ferramentas para edição de imagens que são propostas neste trabalho.

Tabela 2: Ferramentas que serão implementadas

EDIÇÃO	Seletor de área retangular; Recorte retangular; Controle de brilho; Controle de contraste; Rotação para esquerda; Rotação para direita; Redimensionador retangular;
TEXTOS	Seletor de família; Controle de tamanho; Transformação em negrito, itálico, sublinhado e traços; Alinhamento textual; Controle de Cores;
FILTROS	O estúdio disponibilizará pelo menos 2 filtros desenvolvidos pelo autor do projeto para estudo de técnicas em manipulação de imagens.

Para que uma ferramenta possa ser considerada um estúdio ou um editor de imagens, é necessário que possua várias ferramentas para aplicar as edições, caso contrário a ferramenta não passaria de uma biblioteca. O objetivo deste projeto não está em desenvolver apenas uma biblioteca, mas um *framework* que seja útil aos desenvolvedores de sistemas para internet.

23 <http://www.adobe.com/br/products/photoshop.html>

24 <http://www.photoscape.org/ps/main/download.php?lc=pt>

25 <https://instagram.com/>

26 <https://www.befunky.com/>

27 <https://pixlr.com/>

3.2.1 FERRAMENTAS DE EDIÇÃO

As opções de edição consistem em efetuar edições relacionadas ao estado concreto da imagem, ou seja, seu formato, tamanho, posicionamento, aplicação de recortes, entre outras opções. Segue abaixo a descrição sucinta das ferramentas:

- **Seletor e Recorte de área retangular:** O seletor de área retangular consiste em efetuar a seleção de uma área da imagem proporcionando ao usuário as opções de cópia, recorte ou movimento da área selecionada.
- **Controle de Brilho:** A ferramenta de controle de brilho tem como objetivo possibilitar que o usuário aumente ou diminua a porcentagem do brilho da imagem (ou da área selecionada) através de um elemento controlador de intensidade.
- **Controle de Contraste:** A ferramenta de controle de contraste tem como objetivo possibilitar que o usuário aumente ou diminua a porcentagem do contraste da imagem (ou da área selecionada) através de um elemento controlador de intensidade.
- **Rotação para esquerda:** A ferramenta de rotação para esquerda tem como objetivo possibilitar o usuário girar a imagem 90° (noventa graus) para a esquerda da tela do seu dispositivo.
- **Rotação para direita:** A ferramenta de rotação para direita tem como objetivo possibilitar o usuário girar a imagem 90° (noventa graus) para a direita da tela do seu dispositivo.
- **Rotação livre:** A ferramenta de rotação livre vai possibilitar que o usuário gire a imagem para o lado que bem desejar de acordo com a movimentação do mouse.
- **Redimensionador retangular:** A ferramenta de redimensionamento retangular tem como objetivo proporcionar ao usuário o redimensionamento da imagem baseado no movimento do mouse em relação ao centro da imagem.

3.2.2 FERRAMENTAS TEXTUAIS

As ferramentas textuais tem como objetivo proporcionar ao usuário o controle de escritas e estilização das fontes empregadas. Segue abaixo a listagem das ferramentas textuais

que o estúdio ImageJS deverá possuir em sua primeira versão.

- **Aplicação de textos:** Esta ferramenta vai possibilitar que o usuário escreva textos na imagem que está editando.
- **Negrito:** Aplicar e retirar negrito das fontes.
- **Itálico:** Aplicar e retirar itálico das fontes.
- **Sublinhado:** Aplicar e retirar sublinhado as fontes.
- **Controle do Tamanho:** Aumentar ou diminuir o tamanho das fontes;
- **Cores:** Aplicar cores hexadecimais às fontes.
- **Controle da Família:** O usuário poderá aplicar vários estilos de fontes no texto, tais como Arial, Times new Roman, entre outras.

3.2.3 FILTROS

Filtros, é a técnica empregada para alterar as características originais de uma imagem. Como está destacado na Tabela 2, os filtros que serão desenvolvidos no projeto serão de autoria do autor. Não focando apenas na criação de novas texturas, mas implementação de novos algoritmos no processo de manipulação da imagem.

3.3 PROJETAR E DESENVOLVER A INTERFACE

A interface é uma das principais etapas do desenvolvimento do projeto, pois é através dela que o usuário vai interagir com o sistema. Simplicidade e facilidade de uso são de extrema importância para que a ferramenta seja para o usuário uma etapa importante da interação com o sistema, e não apenas um estágio massivo e cansativo.

O desenvolvimento da interface contará com o apoio do *framework front-end* Materialize (seção 2.2.3), pois o mesmo possui sua interface baseada nos estilos do projeto

Material Design desenvolvidos pela empresa Google, a qual já aplicou os princípios primordiais da IHC e continua a amadurecer a ferramenta junto à comunidade.

Tendo em vista que o usuário poderá utilizar a ferramenta através de qualquer dispositivo, a tela do estúdio deverá ser responsiva, ou seja, se adaptar para um *layout* mais apresentável de acordo com o tamanho da tela do dispositivo que o usuário estiver utilizando para acessar. Sendo assim, será necessário planejar o posicionamento dos elementos dispostos em tela. A Figura 6, representa um protótipo de interface para o trabalho proposto.



Figura 6: Protótipo de tela do ImageJS.

Fonte: O autor.

Pode-se observar na Figura 6 que o menu de interação (que contém as opções para edição), está apresentado no formato *dropdown*, pois é o elemento que permite fácil visualização de todas as opções, e possui um comportamento de fácil adaptação para dispositivos móveis. Este protótipo de tela já está desenvolvido com o *framework* Materialize (seção 2.2.3) e não representa todas as opções que serão desenvolvidas.

3.4 PROJETAR E DESENVOLVER FERRAMENTAS

Esta etapa é bastante delicada, pois é aqui que o maior esforço computacional será dedicado. Os algoritmos e bibliotecas que serão utilizados vão definir a qualidade e desempenho da ferramenta em questão.

Para desenvolver as ferramentas que o estúdio possuirá, será utilizado de um recurso conhecido como Canvas²⁸, do HTML5.

O Canvas trata-se de um elemento HTML que em conjunto com a linguagem de programação JavaScript nos permite manipular uma imagem. Sendo possível obter seus *pixels* e trabalhar nestes da maneira que bem entender, seja mudando suas cores, suas posições, entre outras opções.

Caso seja necessário a utilização de uma biblioteca de terceiro para implementar alguma ferramenta, será efetuado uma pesquisa comparativa de relevância entre as bibliotecas em questão. Destacando os itens, usabilidade e consumo de recursos computacionais, como tempo de processamento, memória necessária, entre outros. Algumas destas ferramentas podem ser encontradas no Estado da Arte (seção 2.1).

²⁸ <http://www.w3schools.com/canvas/>

4 DESENVOLVIMENTO

O ImageJS será uma ferramenta *web* apresentada graficamente como um estúdio para edição de imagens. Na sua primeira versão, o objetivo é disponibilizar apenas ferramentas e recursos básicos para aplicar as edições em imagens, possibilitando ampliação de forma a permitir que nas suas versões posteriores sejam desenvolvidas opções mais complexas e de escopo ampliado.

Esta ferramenta será composta por dois módulos distintos. Serão eles o módulo de Editor Incorporado e o módulo do Editor Online.

4.1 EDITOR INCORPORADO

O módulo de Editor Incorporado consiste em disponibilizar o estúdio ImageJS em um sistema *web* qualquer, permitindo o envio de imagens via *web* para sites em qualquer área de atuação.

O Editor Incorporado vai receber uma imagem enviada e logo em seguida enviar pra dentro do estúdio, que por sua vez possuirá as ferramentas para aplicar as edições. Para entender melhor o fluxo de comunicação do ImageJS incorporado com o sistema *web* hospedeiro, foi elaborado um esquema de interação conforme representado na Figura 7.

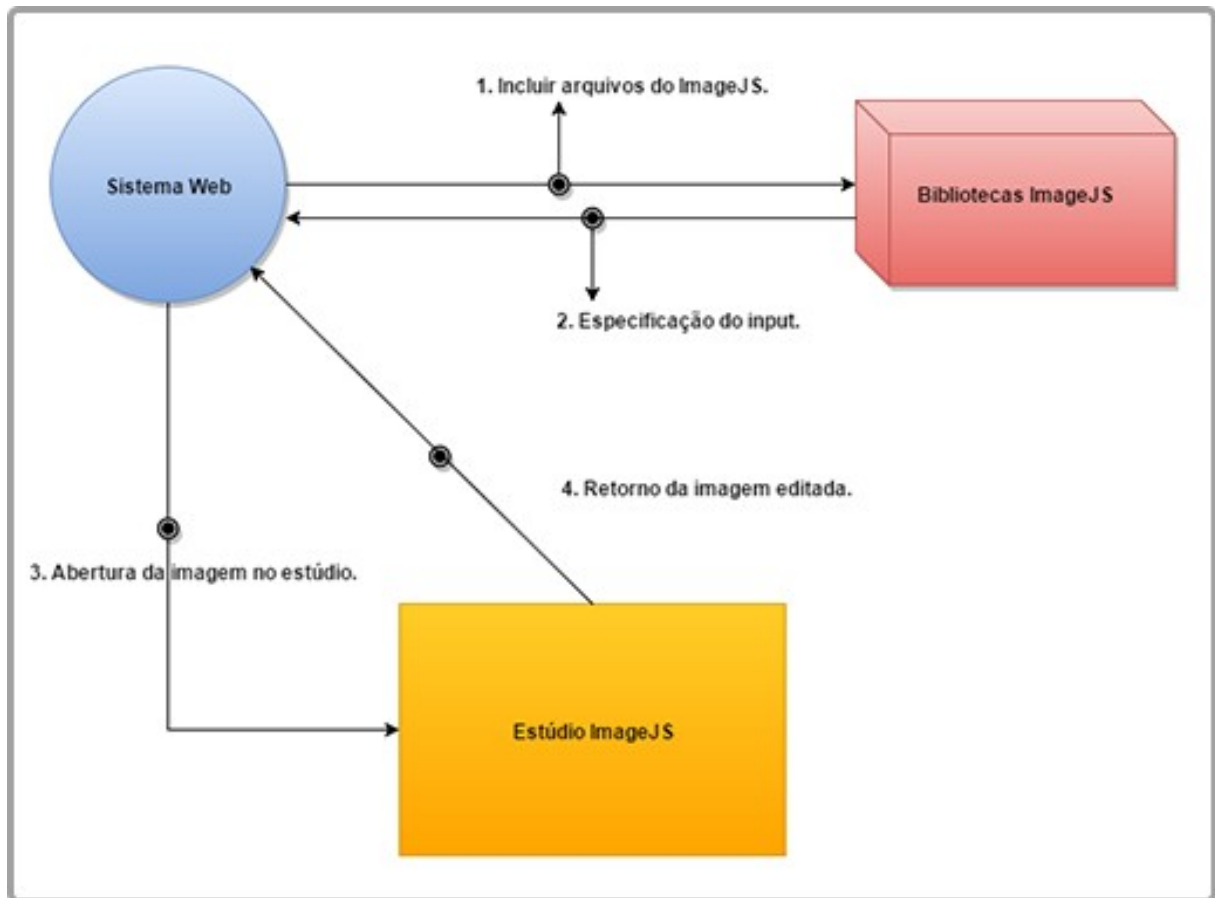


Figura 7: ImageJS Incorporado, esquema de interações.

Fonte: O autor.

Na lista abaixo será explicado brevemente o fluxo grama presente na Figura 7.

1. **Incluir arquivos do ImageJS:** A inclusão dos arquivos ImageJS representam a atribuição lógica das bibliotecas da ferramenta ao sistema *web* hospedeiro. A inclusão deve requerer no corpo HTML da página *web* os arquivos CSS e os arquivos JavaScript. Também é necessário que o sistema terceiro tenha incluído os arquivos referentes ao *framework* jQuery, pois é uma das dependências para o funcionamento correto do estúdio.
2. **Especificação do input:** O elemento HTML que será responsável pelo envio da imagem ao sistema terceiro deve ser um input do tipo *file*²⁹ (arquivo). Como pode haver vários elementos como este em um único documento *web*, devemos informar o

²⁹ http://www.w3schools.com/tags/att_input_accept.asp

ImageJS qual será o input responsável pelo envio da imagem que se deseja editar.

3. **Abertura da imagem no estúdio:** O envio da imagem que será aberta no estúdio, deverá ser feito através do elemento HTML configurado no Item 2. O ImageJS estará monitorando este elemento aguardando pelo envio da imagem, após isso o estúdio será aberto automaticamente com a imagem selecionada para as devidas edições.
4. **Retorno da imagem editada:** A imagem original que foi selecionada ficará armazenada em um espaço de memória referente ao elemento HTML definido no Item 2, a imagem que estará dentro do editor será apenas uma cópia. Após concluir as edições o estúdio vai substituir a imagem original pela cópia editada.

4.2 EDITOR ONLINE

O módulo de Editor Online consiste em disponibilizar o estúdio ImageJS em uma página *web* independente de outros sistemas, para a utilização das ferramentas a qualquer momento.

O Editor online vai disponibilizar opções para selecionar uma imagem a partir da memória de armazenamento do seu dispositivo ou a partir de um endereço *web*. Na primeira versão do estúdio, após concluir as edições será possível apenas que o usuário salve a imagem editada em seu dispositivo. Para entender melhor o fluxo de interação do ImageJS Online, foi elaborado um esquema de interação conforme representa na Figura 8.

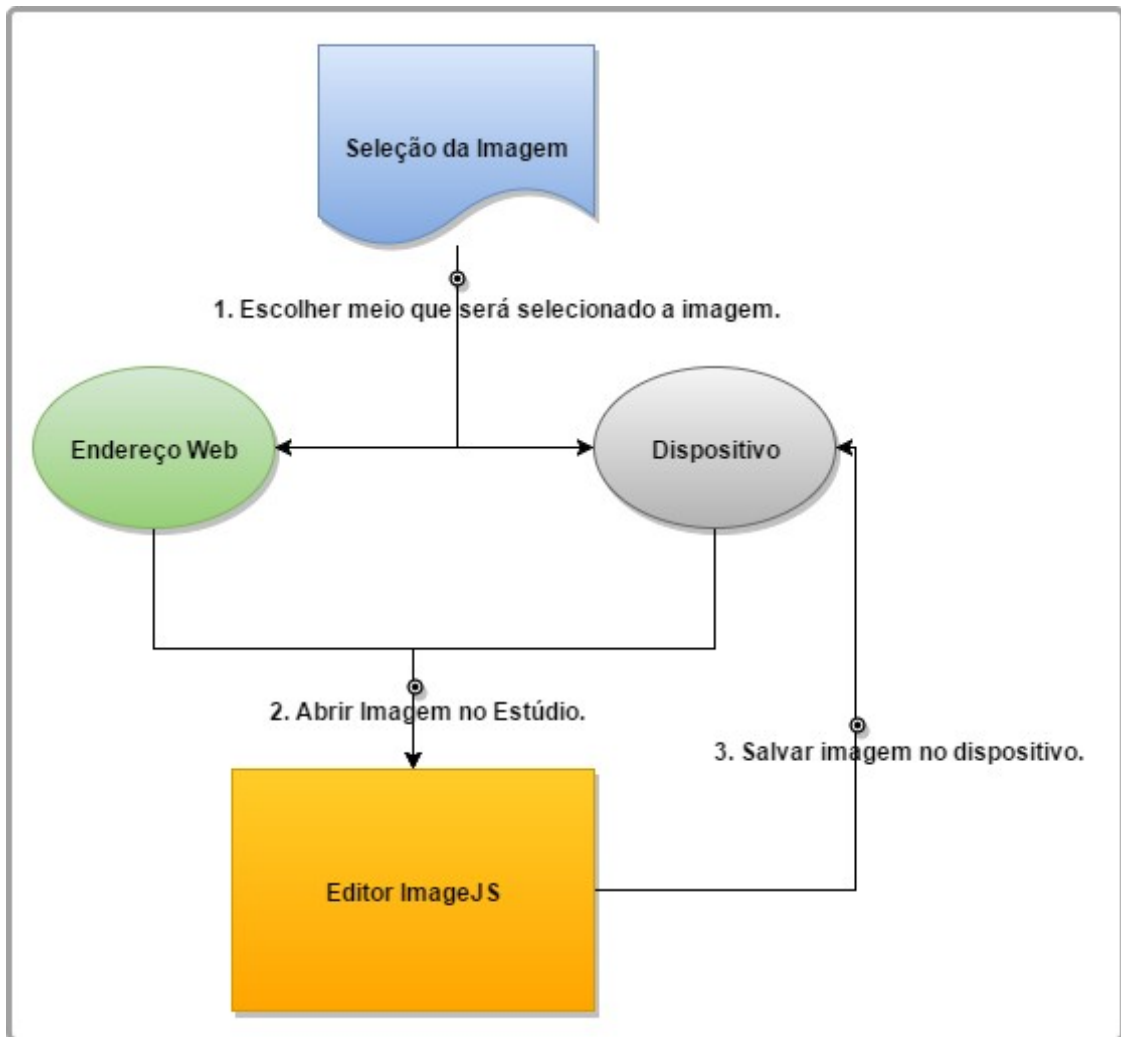


Figura 8: ImageJS Online, esquema de interações.

Fonte: O autor.

Na lista abaixo será explicado brevemente o fluxograma presente na Figura 8.

1. **Escolher meio de seleção da imagem:** Neste passo, o usuário deve selecionar a imagem que será enviada ao estúdio ImageJS. O usuário pode selecionar a imagem diretamente do seu dispositivo de acesso (computador desktop, celular, tablet, entre outros), ou selecionar a imagem de um endereço externo da *web*, informando sua URL.
2. **Abrir imagem no estúdio:** Após o usuário definir o meio de envio da imagem para a ferramenta, o ImageJS vai abrir a mídia no seu estúdio, disponibilizando todas as

ferramentas de edição desenvolvidas disponíveis.

3. **Salvar imagem no dispositivo:** Quando o usuário concluir as edições na imagem que selecionou, o ImageJS disponibilizará a opção para salvar a mídia editada na memória do dispositivo que está utilizando para efetuar as edições.

O estúdio online possuirá uma página inicial responsável por desempenhar o papel de seleção da imagem, conforme mostra a Figura 9:

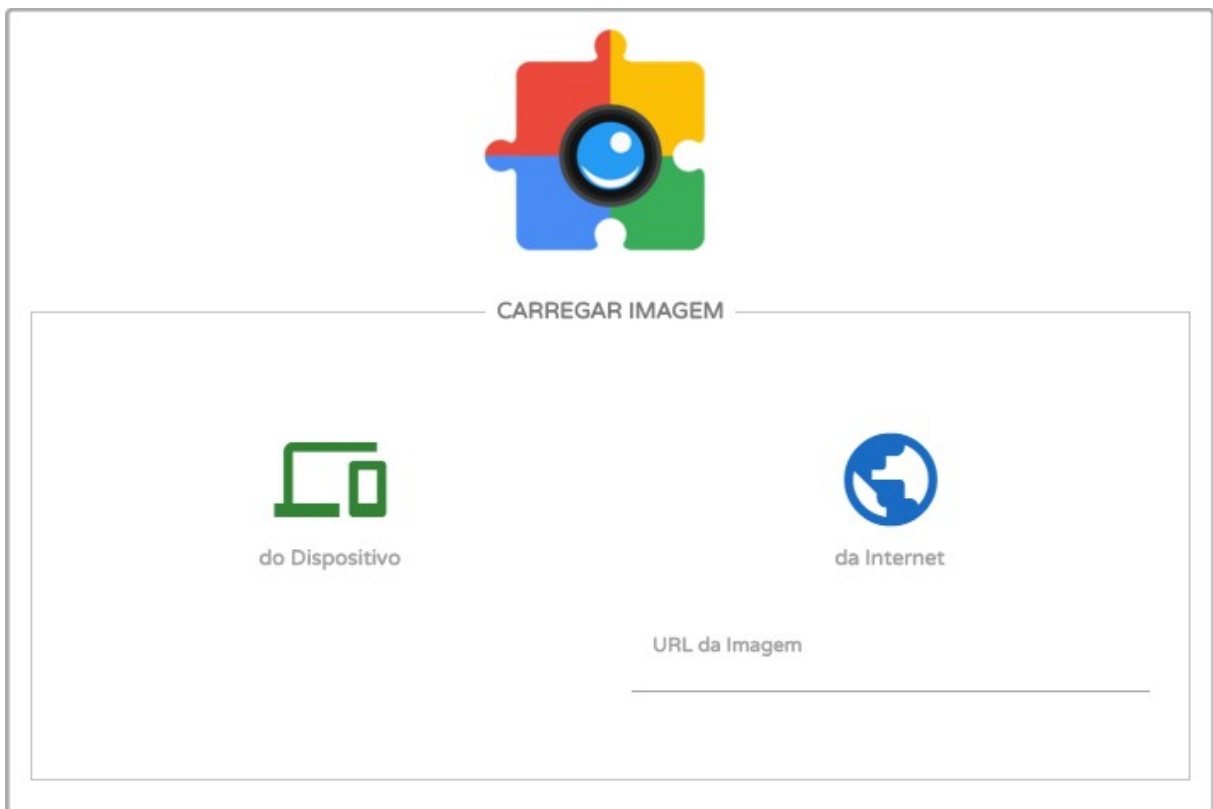


Figura 9: Protótipo de página inicial do ImageJS online.

Fonte: O autor.

Pode-se observar que na Figura 9 as opções de enviar a imagem para o estúdio diretamente da memória do seu dispositivo (através do elemento “do Dispositivo”) ou através de um endereço *web* externo (por meio do elemento “URL da imagem”).

4.3 ARQUITETURA DO SISTEMA

A arquitetura do sistema define como está organizado internamente todos elementos que compõem o sistema. A ferramenta proposta está organizada logicamente em três elementos - Biblioteca de Edição (*framework*), Biblioteca de Estilos, Interface. De acordo com a necessidade dos usuários, os desenvolvedores podem utilizar as partes separadamente. Para entender melhor a organização dos elementos observe a Figura X.X.

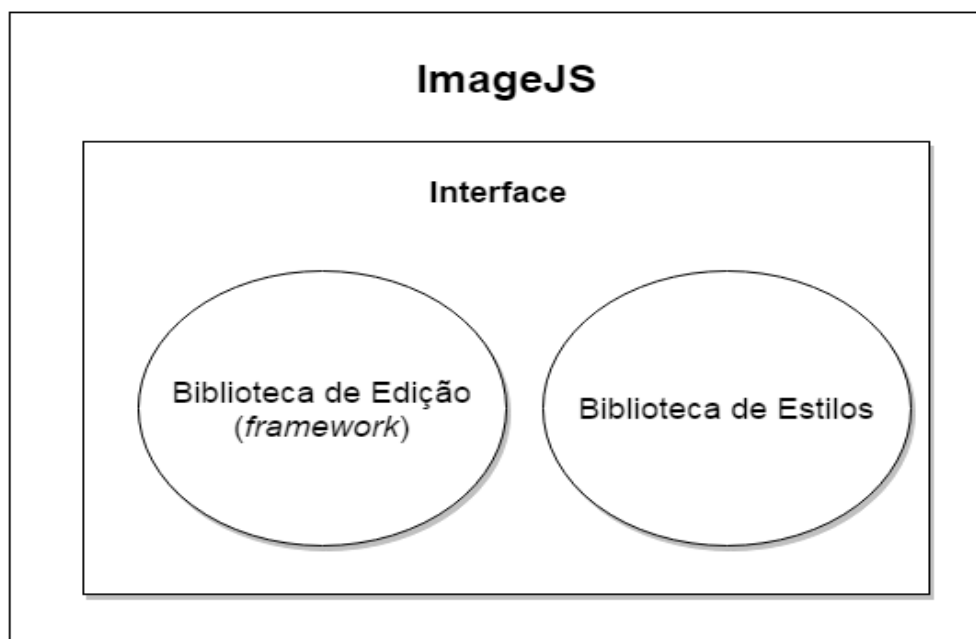


Figura 10: Arquitetura do ImageJS

Fonte: O Autor.

- **Biblioteca de Edição (*framework*):** A biblioteca de edição consiste em uma camada puramente lógica de códigos JavaScript onde todas as funcionalidades e comportamentos responsáveis pelas edições estarão implementados.
- **Biblioteca de Estilos:** A camada de estilos representa todos os códigos CSS que tem como objetivo dar um aspecto visual e comportamental para as tags HTML que estarão presentes na interface.
- **Interface:** A interface consiste nos elementos HTML que serão manipulados pela Biblioteca de Edição e estilizados pela Biblioteca de Estilos. Pode-se dizer também que é a união de atividades dos elementos adjacentes proporcionando todo o aspecto gráfico e funcional da ferramenta.

6 CONSIDERAÇÕES FINAIS

O objetivo deste projeto é propôr uma ferramenta dinâmica de simples usabilidade, que exija poucos recursos computacionais e que seja de fácil acessibilidade para usuários que necessitam acessar através de dispositivos móveis.

Através desta ferramenta, deve ser possível que um usuário comum consiga aplicar ferramentas de edição básica em uma imagem qualquer. O projeto deverá ser cuidadosamente documentado, visando a necessidade da comunidade que possua interesse em cooperar com o contínuo aprimoramento da ferramenta.

A ferramenta, que devido ao seu escopo de aplicabilidade e implementação foi titulada como ImageJS, ou seja, uma união dos termos “imagem” e “JavaScript”; estará apresentada graficamente para o usuário no formato inicial de um mini estúdio para edição de imagens. Onde todas as opções e ferramentas para edição estarão disposta visualmente através de menus. Na tentativa de melhorar a interação com o usuário, estes menus vão se comportar de maneira responsiva, ou seja, assumirão diferentes formas quando acessados por dispositivos que possuem tamanho de tela diferente.

Neste projeto, o principal desafio é identificar quais tecnologias seriam utilizadas para desenvolver as ferramentas de edição que o estúdio disponibilizaria. Após averiguar as bibliotecas já disponíveis e quais recursos poderiam ser utilizados para que fosse possível criar as próprias ferramentas, a tecnologia conhecida como Canvas, disponível a partir do HTML 5, foi a que apresentou maior relevância pois em conjunto com a linguagem de programação JavaScript é possível aplicar técnicas referentes a manipulação de imagem.

Após ser construído as interfaces de interação e os algoritmos de edição, resta unificar os recursos e proporcionar o estúdio em um site online para que seja possível os usuários terem uma pré interação com a ferramenta, antes de incorpora-la em um sistema hospedeiro.

REFERÊNCIAS

ADOBE, Adobe Photoshop CC. <<http://www.adobe.com/br/products/photoshop.html>>. Data de acesso: 12 set.2015.

ALVAREZ, M. **O que é Cross Browser**. <<http://www.criarweb.com/artigos/o-que-e-cross-browser.html>>. Data de acesso: 12 set.2015.

BEFUNKY, Página Inicial. Disponível em <<https://www.befunky.com/create/>>. Data de acesso: 09.nov.2015.

CRANE, D. et al. **Ajax em ação**. 1. ed. Londres: Manning Publications, 2006.

FAYAD, M. E. et al. **Object-oriented Application frameworks**. **Communications of the ACM**, Vol. 40, 10 p., 1997.

GOOGLE, Material Design. Disponível em <<https://www.google.com/design/spec/components/tabs.html>>. Data de acesso: 09.nov.2015.

IMAZEN. StudioJS. <<http://studio.imageresizing.net/studio.html>>. Data de acesso: 12 set.2015.

LEFEVRE, R. Página Inicial. <<http://camanjs.com/>>. Data de acesso: 12 set.2015.

LUCA, C. Buscas no Google: 15% das pesquisas diárias são novas, nunca foram feitas. Disponível em: <<http://idgnow.com.br/internet/2013/09/25/busca-google-15-das-pesquisas-diarias-nunca-foram-feitas-antes/>>. Data de acesso: 12 set.2015.

MARTINS, Pedro. O que é front-end ?. <<http://mobgeek.com.br/blog/o-que-e-front-end>>. Data de acesso: 10.nov.2015.

MAZZA, L. **HTML5 e CSS3: Domine a web do futuro**. Casa do código, 2013. ISBN

9788566250053. Disponível em: <<http://www.casadocodigo.com.br/products/livro-html-css>>.

MOZILA, Canvas tutorial. <https://developer.mozilla.org/pt-BR/docs/Web/Guide/HTML/Canvas_tutorial>. Data de acesso: 10.nov.2015.

PhotoScape, Página Inicial. <<http://www.photoscape.org/ps/main/index.php>>. Data de acesso: 12 set.2015.

PIXLR. Express. <<https://pixlr.com/>>. Data de acesso: 12 set.2015.

RAMALHO, J. A. **Curso completo para desenvolvimento web**. Rio de Janeiro: Elsevier, 2005.

REMOALDO, P. **O Guia prático do Dreamweaver CS3 com PHP, JavaScript e Ajax**. 1. ed. Lisboa: Centro Atlântico, 2008.

RUTTER, J. **Smashing jQuery interatividade avançada com JavaScript simples**. Porto Alegre: Bookman Companhia Editora Ltda, 2012

SILVA, M. S. **Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3**. São Paulo: Novatec Editora, 2012.

The Gimp Team. Gimp. <<http://www.gimp.org/>> Data de acesso: 12 set.2015.