

Nelia Cantanhede Reis

Matrícula: 2012395

Sistema de Marcação em Imagens 2D utilizando Fabric.js

Rio de Janeiro - RJ

Julho - 2021

Sumário

1	INTRODUÇÃO	2
1.1	Motivação	3
1.2	Objetivo	3
1.3	Organização do Documento	3
2	FUNDAMENTAÇÃO TEÓRICA	4
2.1	A imagem digital	4
2.2	Formatos de arquivos	4
2.3	Arquivo JSON	5
2.4	Fabric.js	5
2.5	Entidades geométricas	5
2.5.1	Ponto	5
2.5.2	Reta	6
3	REQUISITOS E ESPECIFICAÇÕES	7
3.1	Requisitos funcionais e Não funcionais	7
3.1.1	Requisitos funcionais	7
3.1.2	Requisitos não funcionais	9
3.2	Casos de Uso	9
4	DESENVOLVIMENTO	13
4.1	Plataformas e tecnologias	13
4.2	Diagrama de classes	13
5	TESTES	15
5.1	Testes dos requisitos	15
6	MANUAL DO USUÁRIO	17
6.1	Requisitos mínimos	17
6.2	Principais funções do programa	17
	REFERÊNCIAS	21

1 Introdução

Com a crescente popularidade da rede global de computadores nas mais diferentes áreas da atividade humana, a Internet vem sendo largamente utilizada como um eficiente meio de busca e troca de informações ([SOSHNIKOV, 1998](#)), fornecimento e utilizações de variados tipos de aplicações Web. Um sistema web é um sistema online associado a um recurso web, normalmente uma página na Internet. Com um sistema de anotação, o utilizador pode adicionar, modificar ou remover marcações de uma imagem através da página web sem modificar sua própria imagem original ([NEVES et al., 2010](#)).

O sistema de marcação desenvolvido serve para os especialistas fazerem marcações em imagens digitais, com intuito de informar anomalias nelas, caso haja. Com isso, fica mais fácil outras pessoas interpretarem essas imagens, e assim utilizá-las para realizar estudos nas mais diversas áreas.

Existem ferramentas que são desenvolvidas para auxiliar nas marcações em imagens através da web, a *Fabric.js* é uma delas. Se trata de uma biblioteca *javascript* poderosa e bastante simples para telas *HTML5*. Ele fornece uma plataforma interativa para trabalhar com a tela *HTML5*. Usando a *Fabric*, você pode criar objetos e formas na tela de formas geométricas simples a formas complexas, ela também oferece suporte a animações. Fornece controles para trabalhar com imagens, como arrastá-las, dimensioná-las e girá-las. Ela permite o agrupamento de formas e objetos que podem ser manipulados juntos. Suporta eventos não apenas limitados ao elemento *canvas*, mas também com objetos e ainda fornece funcionalidade para serializar a tela em SVG ou JSON e reutilizá-la como e quando necessário([FABRIC.JS, 2013](#)).

O presente trabalho apresenta uma aplicação web que permite que seus usuários realizem marcações em imagens 2D. Para realizar esse processo será utilizado a biblioteca *Fabric.js*.

1.1 Motivação

Existem sistemas que facilitam a criação de anotações, mas são de arquiteturas stand alone, ou seja, necessitam de um executável instalado na máquina cliente. Por isso, eles possuem limitações de compatibilidade de plataformas e tem a necessidade de instalação de aplicativos e plugins nas máquinas de seus usuários (SERIQUE, 2012).

Podemos ver que existe uma carência por ferramentas web de criação de anotações em imagens, sendo assim a principal motivação deste trabalho é a possibilidade de desenvolver uma ferramenta web de marcação que possa realizar os requisitos atendidos pelos atuais sistemas de anotações de imagens *desktop*.

1.2 Objetivo

Desenvolver um sistema de marcação web para imagens 2D, visando auxiliar os especialistas a realizarem marcações em imagens de forma mais rápida e prática.

1.3 Organização do Documento

A organização do trabalho é realizada da seguinte forma: No Capítulo 2 temos a Fundamentação teórica onde, contém conceitos fundamentais para o contexto e entendimento do trabalho. O Capítulo 3 apresenta os Requisitos e Especificações, mostra-se os Requisitos Funcionais e Não Funcionais coletados do sistema e o diagrama de Casos de Uso contendo a descrição e pré-condições de cada caso.

No Capítulo 4, mostra o Desenvolvimento, apresenta-se as plataformas e tecnologias utilizadas para desenvolver o sistema, bem como seu diagrama de Classes e diagramas de sequência para melhor entendimento de funcionalidades. O Capítulo 5, mostra casos de testes que foram realizados no sistema. No último Capítulo o 6, temos o Manual do Usuário, contendo explicações de como usar o programa corretamente.

2 Fundamentação teórica

Neste capítulo são discutidos os principais fundamentos da imagem digital e técnicas computacionais aplicados no trabalho.

2.1 A imagem digital

A tecnologia da imagem digital propicia os meios de se codificar digitalmente documentos analógicos em forma de imagem digital para armazenagem, transmissão e recuperação em sistemas computadorizados ([WATERS, 2003](#)).

As imagens digitais podem ser produzidas por diversos aparelhos, incluindo câmeras de vídeo e fotográficas digitais, scanners, aparelhos de raios-X, microscópios eletrônicos, aparelhos de ultra-som e radares ([ASKEY, 2003](#)). Desta forma, deve-se diferenciar a terminologia "imagem digital" de "fotografia digital". Esta última é somente um tipo de imagem digital que é adquirida com o uso de câmeras fotográficas digitais.

Como os computadores utilizam um sistema de numeração binária, ou seja, apenas dois dígitos para representar qualquer caractere, seja este, um número, uma letra ou uma fotografia, as imagens digitais, bem como todos os arquivos digitais, nada mais são do que uma série de dígitos binários: 0 e 1. Ou seja, a imagem é uma longa sequência destes dois números que, então, se "transformam" em quadros minúsculos. Cada pequeno quadro, denominado pixel (contração de picture X element, ou elementos da imagem), representa uma intensidade de luz e de cor que todos estes, em conjunto, irão formar a imagem digital ([GPD, 2003](#)). Logo, a unidade básica formadora da imagem digital é o pixel.

2.2 Formatos de arquivos

O formato de arquivo é a forma como a imagem será armazenada, isto é, guardada ou salva no scanner, ou na câmera fotográfica digital, e posteriormente transferida para o computador. Existem inúmeros formatos, porém os mais utilizados para a obtenção de imagens digitais são o JPEG (ou .jpg) e o PNG (ou .png) ([ABELSON, 1999](#)).

O formato JPEG (Joint Photographic Experts Group) é o mais popular, pois é rápido e fácil de trabalhar nos diversos programas de edição de imagens, além de ocupar pouco espaço (em bytes) no disco rígido do computador. A grande desvantagem desse formato é que a qualidade final da imagem é diminuída através de sua compressão, ou seja, alguns pixels são descartados (eliminados). O formato PNG (Portable Network Graphics) ela foi criada para facilitar a troca de imagens pela internet. Assim como o .jpg, esse

formato usa compressão para reduzir o peso dos arquivos; diferentemente de .jpg, no entanto, o tipo de compressão que o .png usa não implica tanta perda de qualidade especialmente no caso de arquivos gráficos ([REDAÇÃO, 2018](#)).

2.3 Arquivo JSON

JSON é um formato de texto derivado da linguagem Javascript e é representado por uma coleção de objetos compostos por pares de chave-valor conforme definição da 8 a edição do ECMAScript Language Standard ([ECMA, 1999](#)), que é responsável pela definição da linguagem Javascript.

Segundo [Bray et al. \(2014\)](#), o JSON é uma estrutura de texto no formato de objeto Javascript e pode ser composto de subobjetos. Sua estrutura possibilita uma diversificada representação de dados, pois o mesmo suporta quatro tipos de dados primitivos e estrutura de subobjetos, como strings, number, integer, boolean, object e array.

2.4 Fabric.js

Se trata de uma biblioteca javascript poderosa e bastante simples para telas HTML5. Ele fornece uma plataforma interativa para trabalhar com a tela HTML5. Usando a Fabric, você pode criar objetos e formas geométricas simples a formas complexas na tela, ela também oferece suporte a animações. Fornece controles para trabalhar com imagens, como arrastá-las, dimensioná-las e girá-las. Ela permite o agrupamento de formas e objetos que podem ser manipulados juntos. Suporta eventos não apenas limitados ao elemento canvas, mas também com objetos e ainda fornece funcionalidade para serializar a tela em SVG ou JSON e reutilizá-la como e quando necessário ([FABRIC.JS, 2013](#)).

2.5 Entidades geométricas

2.5.1 Ponto

O ponto é um objeto que não possui definição, dimensão e forma. Por isso, é impossível encontrar qualquer medida nele, como comprimento, largura, altura, área, volume etc. O ponto é a base de toda a Geometria, pois é a partir de conjuntos deles que são formadas as figuras geométricas. Os pontos são usados para representar localizações no espaço. Como não possuem tamanho ou forma, uma localização em algum espaço fica bem definida quando está em algum ponto ([SILVA, 2019](#)).

2.5.2 Reta

São conjuntos de pontos compreendidos como linhas infinitas que não fazem curvas. Embora sejam formadas por pontos, também não possuem definição, mas apenas essa característica. Obviamente, são necessários infinitos pontos para construir uma reta. É possível medir a distância entre dois pontos específicos que estão sobre uma reta. Entretanto, continua não sendo possível medir a largura da reta, pois os pontos que a formam não possuem dimensões. Por essa razão, dizemos que a reta é um objeto unidimensional, ou seja, que possui uma única dimensão ([SILVA, 2019](#)).

3 Requisitos e Especificações

Neste capítulo apresentados os principais requisitos e especificações para a criação do sistema de marcação de imagens.

3.1 Requisitos funcionais e Não funcionais

Os requisitos serão classificados em relação a prioridade, que pode ser avaliado como: essencial, importante ou desejável e em relação ao esforço que pode ser baixo, médio ou alto. Essa classificação é importante para estimar o tempo de modelagem e desenvolvimento.

3.1.1 Requisitos funcionais

O sistema é constituído dos seguintes requisitos funcionais descritos abaixo:

1. Leitura e carregamento de uma imagem.

Descrição	O usuário poderá abrir uma imagem selecionando e carregando para o programa
Prioridade	Essencial
Esforço	Alto

2. Leitura e carregamento de um arquivo JSON.

Descrição	O usuário poderá abrir um arquivo json, que já contenha marcações selecionando e carregando para o programa.
Prioridade	Essencial
Esforço	Alto

3. Salvamento no formato PNG

Descrição	O usuário após realizar marcações, poderá salvar elas como imagem no formato PNG
Prioridade	Desejável
Esforço	Baixo

4. Salvamento no formato JSON

Descrição	O usuário após realizar marcações, poderá salvar as marcações como um arquivo json, para posterior edição.
Prioridade	Essencial
Esforço	Alto

5. Marcação livre

Descrição	O usuário poderá realizar marcações a mão livre.
Prioridade	Desejável
Esforço	Baixo

6. Marcação direta

Descrição	O usuário poderá realizar marcações direta, utilizando pontos e linhas.
Prioridade	Essencial
Esforço	Alto

7. Edição das marcações

Descrição	Uma vez que o usuário realizou marcações nas imagens, ou tenha carregado um arquivo json com marcações contidas nele, ele pode editar as marcações livremente.
Prioridade	Essencial
Esforço	Alto

8. Limpar a tela

Descrição	O usuário poderá limpar a tela a qualquer momento para começar novas marcações.
Prioridade	Importante
Esforço	Médio

3.1.2 Requisitos não funcionais

O sistema é constituído dos seguintes requisitos não funcionais descritos abaixo:

1. Portabilidade

Descrição	O sistema poderá ser executado em diferentes sistemas operacionais, tais como Linux e Windows, necessitando somente de um browser.
Prioridade	Desejável
Esforço	Médio

2. Robustez

Descrição	O sistema deverá ser estável, ou seja, não deverá travar ou fechar inesperadamente.
Prioridade	Essencial
Esforço	Alto

3. Facilidade de uso

Descrição	O sistema deve ser intuitivo e de fácil manuseio.
Prioridade	Desejável
Esforço	Baixo

3.2 Casos de Uso

Nessa seção é apresentado o diagrama de casos de uso criado para o sistema. A Figura 1 mostra a criação dele com base nos requisitos do sistema, assim como a especificação de cada um.

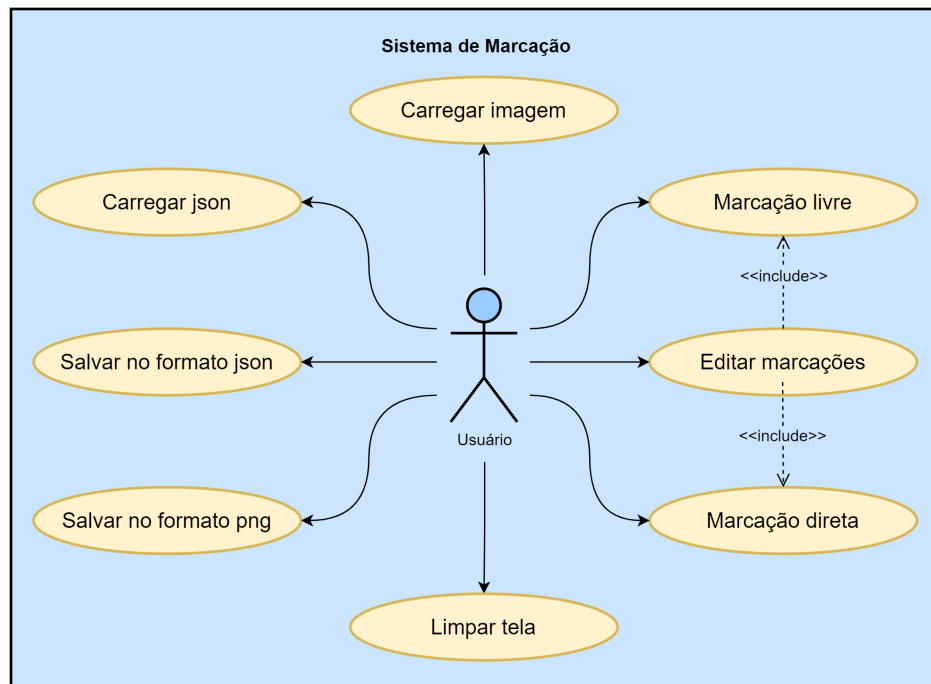


Figura 1 – Diagrama de casos de uso.

1. Carregar imagem.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Abrir e carregar uma imagem.
Pré-condições:	Estar com o sistema aberto
Pós-condições:	Exibir a imagem carregada na tela.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário escolhe a opção carregar imagem; 3. O sistema mostra a imagem na tela;
Cenário alternativo:	Se a imagem for inválida ou estiver corrompida o sistema não abrirá ela.

2. Carregar json.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Abrir e carregar um arquivo json que contém marcações.
Pré-condições:	Estar com o sistema aberto
Pós-condições:	Mostrar as marcações que foram carregadas na tela.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário escolhe a opção carregar json; 3. O sistema mostra as marcações carregadas na tela;
Cenário alternativo:	Se o arquivo tiver outra extensão, diferente de json o sistema não abrirá ele.

3. Salvar no formato json.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Depois que o usuário realizar marcações ele escolhe esta opção para salvá-las em um json.
Pré-condições:	Estar com o sistema aberto e ter realizado pelo menos uma marcação.
Pós-condições:	Ter realizado o download na máquina do usuário.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário realiza marcações; 3. O usuário escolhe a opção de salvar como json; 4. O sistema realiza o download na máquina do usuário;
Cenário alternativo:	Não ter marcações para serem baixadas.

4. Salvar no formato png.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Depois que o usuário realizar marcações ele escolhe esta opção para salvá-las no formato png.
Pré-condições:	Estar com o sistema aberto e pode ou não ter realizado marcações.
Pós-condições:	Ter realizado o download na máquina do usuário.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário realiza marcações, ou não; 3. O usuário escolhe a opção de salvar no formato png; 4. O sistema realiza o download na máquina do usuário;
Cenário alternativo:	Não ter nada a ser salvo.

5. Marcação Livre.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Opção para o usuário realizar marcações a mão livre.
Pré-condições:	Estar com o sistema aberto.
Pós-condições:	Exibir as marcações que foram realizadas na tela.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário escolhe a opção marcação livre; 3. O usuário realiza as marcações na tela; 4. O sistema exibe as marcações que foram feitas;
Cenário alternativo:	Não exibir nada, caso o usuário não tenha escolhido a opção realizar marcação livre.

6. Marcação Direta.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Opção para o usuário realizar marcações direta, utilizando pontos e linhas.
Pré-condições:	Estar com o sistema aberto.
Pós-condições:	Exibir as marcações que foram realizadas na tela.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário escolhe a opção marcação direta; 3. O usuário realiza as marcações na tela; 4. O sistema exibe as marcações que foram feitas;
Cenário alternativo:	Não exibir nada, caso o usuário não tenha escolhido a opção realizar marcação direta.

7. Editar Marcações.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Uma vez que o usuário realizou marcações nas imagens, ou tenha carregado um arquivo json com marcações contidas nele, ele pode editar as marcações livremente.
Pré-condições:	Estar com o sistema aberto e ter realizado marcações na tela ou carregado elas a partir de uma arquivo json.
Pós-condições:	Exibir as edições que foram realizadas na tela.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. O usuário escolhe a opção editar; 3. Tendo marcações na tela, o usuário pode editar livremente as marcações; 4. O sistema exibe as edições que foram feitas;
Cenário alternativo:	Se não tiver nenhuma marcação na tela, nada acontecerá quando escolher essa opção.

8. Limpar Tela.

Descrição	
Ator:	O usuário.
Descrição sucinta:	Serve para apagar as marcações da tela
Pré-condições:	Estar com o sistema aberto e ter realizado marcações na tela ou carregado elas a partir de uma arquivo json, assim poderá apagá-las.
Pós-condições:	Exibir a tela limpa.
Cenário principal:	<ol style="list-style-type: none"> 1. O usuário acessa a página da aplicação; 2. Se tiver marcações na tela; 3. O usuário escolhe a opção limpar; 4. O sistema exibe a tela limpa;
Cenário alternativo:	Se não tiver nenhuma marcação na tela, nada acontecerá quando escolher essa opção.

4 Desenvolvimento

Esta seção apresenta informações sobre a fase de desenvolvimento da aplicação, como a biblioteca, ferramentas utilizadas e o diagrama de classes.

4.1 Plataformas e tecnologias

O programa foi desenvolvido utilizando a linguagem javascript, com a biblioteca Fabric.js, foi utilizada a IDE Visual Studio Code para dar suporte ao desenvolvimento. Utilizou-se também HTML 5 com o uso de css para dar um estilo as páginas do sistema. A tabela 1 resume as plataformas e tecnologias utilizadas no desenvolvimento do programa.

Tabela 1 – Plataformas e tecnologias utilizadas.

Linguagem de programação:	Javascript
IDE:	Visual Studio Code
Biblioteca:	Fabric.js
Plataformas:	Windows, Linux, Mac

4.2 Diagrama de classes

A figura 2 apresenta o diagrama de classes desenvolvido para o sistema.

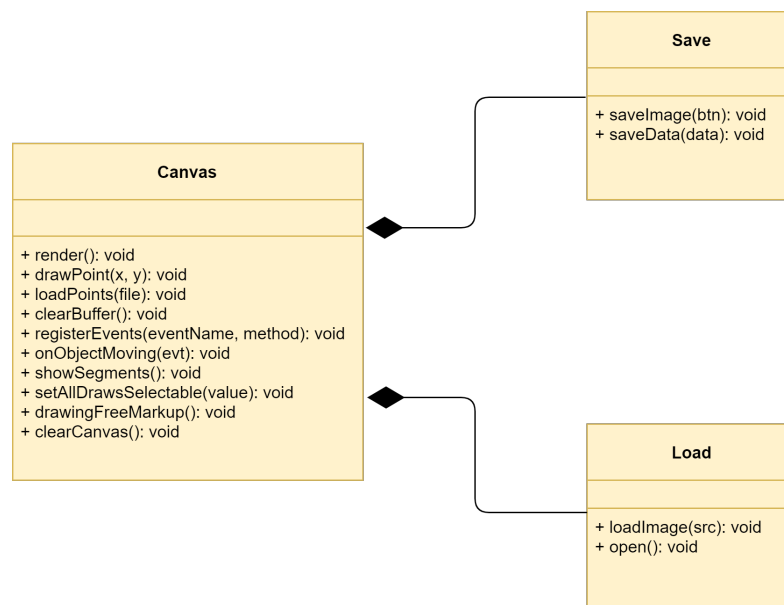


Figura 2 – Diagrama de classes do sistema.

A classe Canvas contém quase todos os métodos que são necessários para o funcionamento do sistema, é através dela que todas as funcionalidades realizadas na tela são executadas. Foram criadas mais duas classes secundárias, a classe Save que serve exclusivamente para realizar o salvamento das marcações, tanto no formato png, quanto no formato json e a classe Load que é responsável por abrir e carregar os arquivos, elas compõem a classe Canvas, ou seja, dependem diretamente dessa classe primária para funcionarem.

5 Testes

Este capítulo apresenta os testes realizados para garantir a funcionalidade do sistema.

5.1 Testes dos requisitos

Foram realizados testes manuais para verificar o comportamento do sistema. Foi feito um roteiro para cada um dos requisitos funcionais mencionados na seção 3.1.

1. Leitura e carregamento de uma imagem, permite o usuário carregar uma imagem 2d. A Figura 3 mostra o resultado obtido.

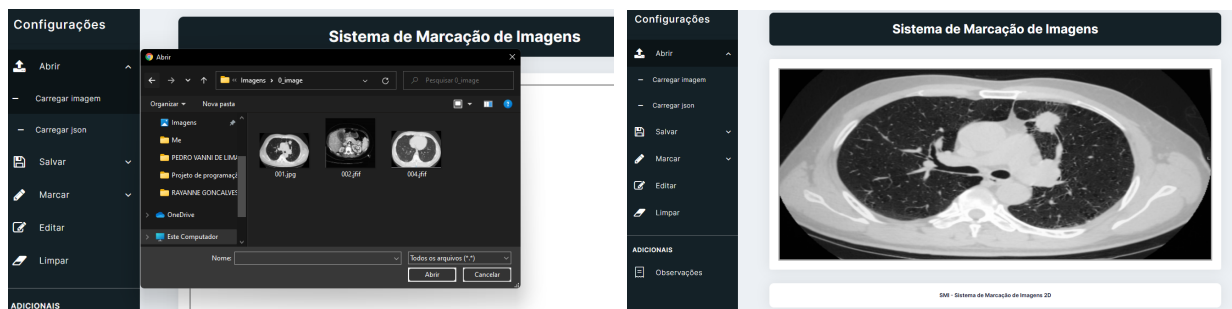


Figura 3 – Resultado do primeiro caso de teste, carregamento de uma imagem.

2. Leitura e carregamento de um arquivo JSON, permite o usuário carregar um arquivo json. A Figura 4 mostra o resultado obtido.

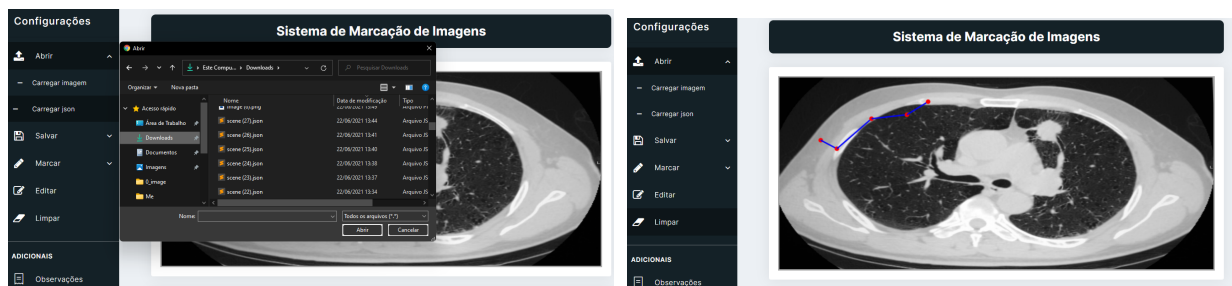


Figura 4 – Resultado do segundo caso de teste, carregamento de um arquivo json.

3. Realizar marcações, o usuário pode optar por fazer marcações livres ou diretas, ou ambas ao mesmo tempo. A Figura 5 mostra o resultado obtido do teste.

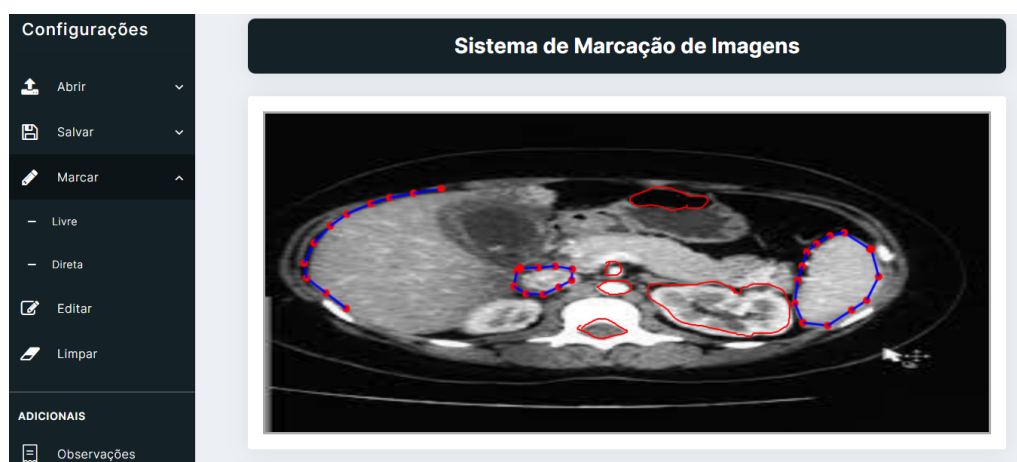


Figura 5 – Resultado do terceiro caso de teste, marcações livre e direta.

4. Edição de marcações, permite o usuário editar as marcações que já existem livremente. A Figura 12 mostra o resultado do teste obtido.



Figura 6 – Resultado do quarto caso de teste, edição de marcações. A esquerda imagem marcada, a direita mesma imagem editada.

5. Limpar tela, permite o usuário limpar as marcações que ele realizou na imagem. A Figura 7 mostra o resultado do teste obtido.

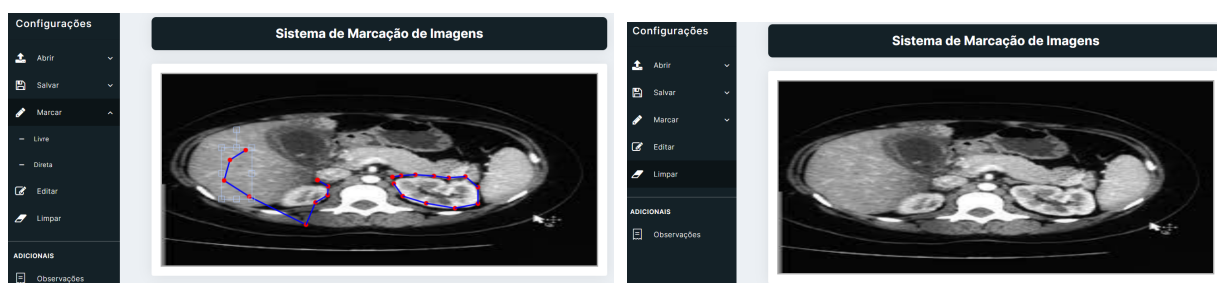


Figura 7 – Resultado do quinto caso de teste, limpar tela. A esquerda imagem marcada, a direita a tela limpa.

6 Manual do Usuário

Esta seção apresenta o manual do usuário. Nela, é demonstrado como utilizar o sistema de marcação de imagens.

6.1 Requisitos mínimos

O sistema a princípio está funcionando localmente, basta o usuário possuir o sistema em seu computador e ter um navegador para testá-lo.

6.2 Principais funções do programa

Para o usuário abrir uma imagem ou um arquivo json, primeiramente ele deve selecionar a opção ‘Abrir’, em seguida escolher entre as duas opções disponíveis que são: ‘Carregar imagem’ ou ‘Carregar json’. Após isso, a janela de seleção de arquivo será exibida, tal como na Figura 8.



Figura 8 – Janela de seleção de arquivo.

Se o usuário optou por escolher uma imagem, ela será carregada na tela e o mesmo poderá começar a realizar as marcações desejadas, basta ele selecionar a opção ‘Marcar’, nela contém também duas opções de marcações: ‘Livre’ e ‘Direta’. A Figura 9 exibe essas opções mencionadas.



Figura 9 – Opções de marcações Livre e Direta.

Se escolher marcação livre o usuário deverá posicionar o ponteiro do mouse em qualquer parte da imagem para começar a marcação, ele deve clicar e segurar o botão esquerdo do mouse e arrastá-lo para onde deseja marcar, uma linha vermelha aparecerá acompanhando o mouse, para terminar de marcar basta soltar o botão. A Figura 10 apresenta um exemplo de marcação livre.

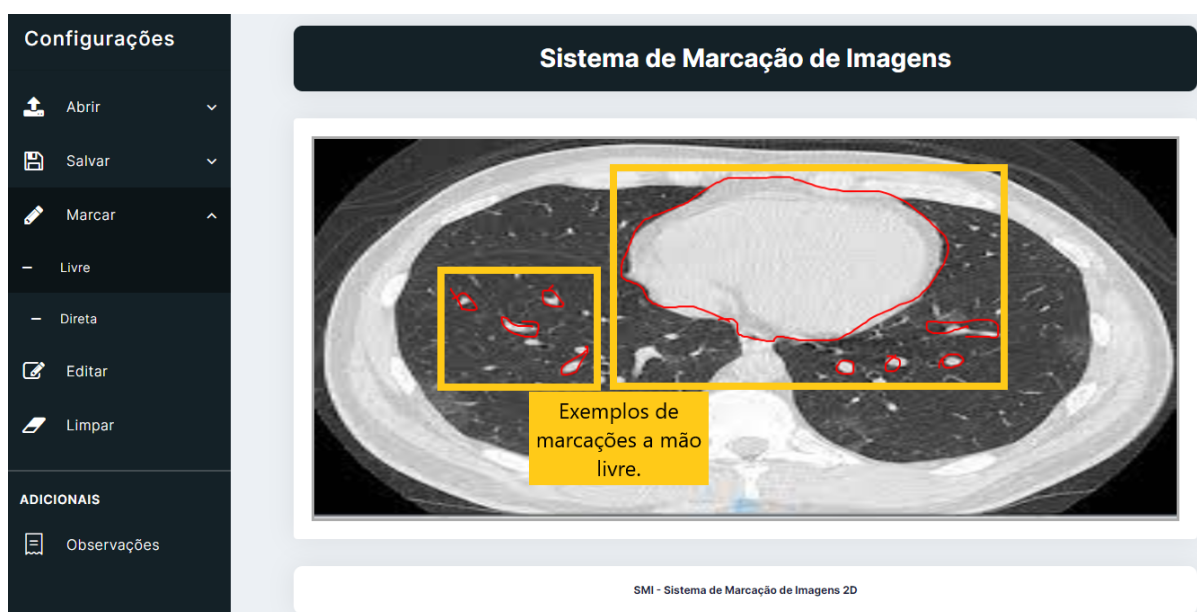


Figura 10 – Exemplo de marcação livre.

Já na marcação direta, o usuário deve apertar o botão esquerdo do mouse e soltá-lo, irá aparecer um pontinho vermelho, assim ele deve escolher outro local na tela e clicar

novamente, aparecerá uma linha ligando os dois pontos inicialmente desenhados, assim a medida que ele coloca os pontos as linhas vão se conectando, para terminar a marcação basta dar duplo click com o botão esquerdo do mouse. A Figura 11 apresenta um exemplo de marcação direta.

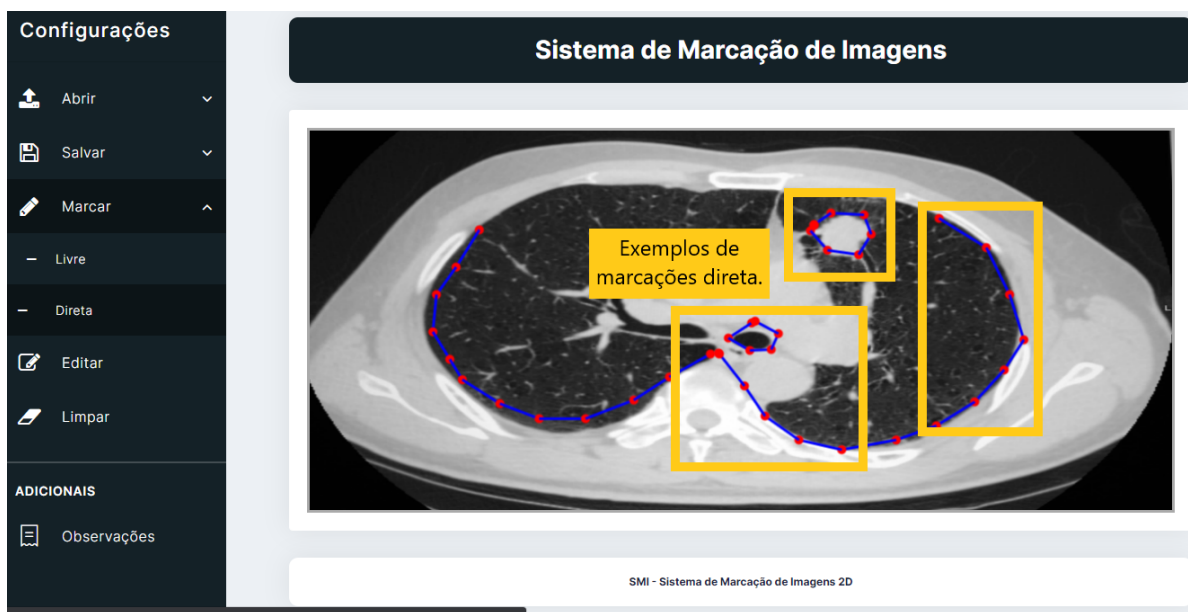


Figura 11 – Exemplo de marcação direta.

Para realizar a edição das marcações o usuário deve selecionar a opção 'Editar', ele pode editar marcações que já estejam na tela ou carregar um arquivo json que contém marcações. O usuário deve clicar e segurar com o botão esquerdo do mouse em um dos pontos que fazem parte da marcação, e arrastá-lo para onde deseja movê-lo. Ele pode selecionar vários pontos e fazer o mesmo procedimento. A Figura 12 mostra um exemplo de edição feito a partir da Figura 11

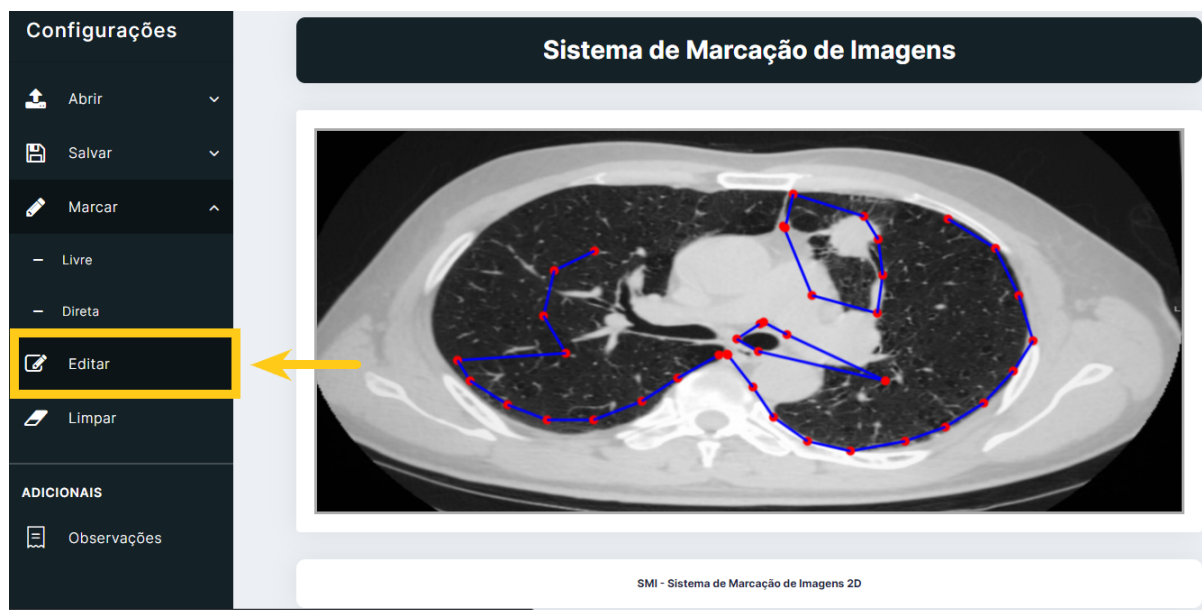


Figura 12 – Exemplo de uma edição de marcação.

Para apagar as marcações realizadas, o usuário deve selecionar a opção ‘Limpar’, assim todas elas serão apagadas da tela. A Figura 13 apresenta o resultado de limpeza de tela, usado nas marcações da Figura 12

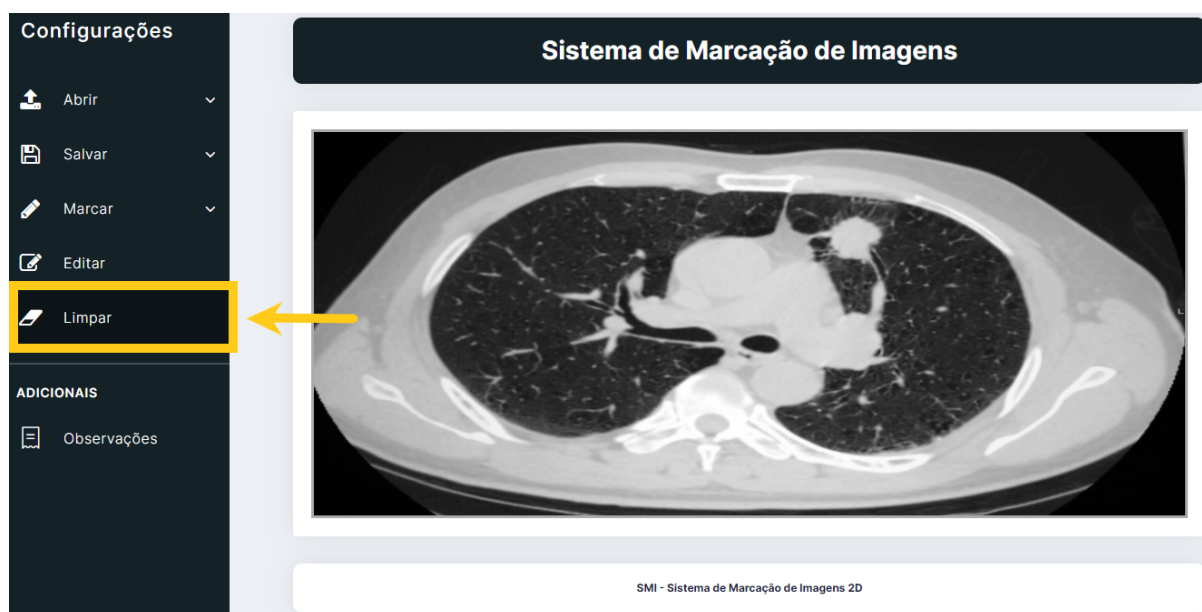


Figura 13 – Tela sem marcações, após o uso da opção Limpar.

O usuário pode fazer o download de suas marcações, basta selecionar a opção ‘Salvar’, nela contém as opções de ‘Salvar como PNG’ ou ‘Salvar como JSON’, após escolher uma dessas opções, será feito o download em seu computador.

Referências

- ABELSON, M. Digital imaging update. *American journal of orthodontics and dentofacial orthopedics*, Elsevier, v. 116, n. 5, p. 587–590, 1999. Citado na página 4.
- ASKEY, P. *Learn: glossary: digital imaging: RAW image format*. 2003. Citado na página 4.
- BRAY, T. et al. The javascript object notation (json) data interchange format. RFC 7159, DOI 10.17487/RFC7159, March 2014, < <http://www.rfc-editor.org> ... , 2014. Citado na página 5.
- ECMA, F. D. Ecma script language specification. 1999. Citado na página 5.
- FABRIC.JS. *Fabric.js is a powerful and simple Javascript HTML5 canvas library*. 2013. Disponível em: <<http://fabricjs.com/>>. Citado 2 vezes nas páginas 2 e 5.
- GPD. *GUIA prático digital: um mundo em evolução*. [S.l.]: São Paulo, 2003. 10–13 p. Citado na página 4.
- NEVES, L. A. d. S. R. d. et al. Sistema web semi-automático para anotação colaborativa de média h. 264 e flash vídeo. 2010. Citado na página 2.
- REDAÇÃO, O. *JPG, PNG, GIF e BMP – quais as diferenças entre os principais formatos de imagens?* 2018. Accessed: 29-06-2021. Disponível em: <<https://olhardigital.com.br/2017/06/08/dicas-e-tutoriais/jpg-png-gif-e-bmp-quais-as-diferencas-entre-os-principais-formatos-de-imagens/>>. Citado na página 5.
- SERIQUE, K. J. d. A. *Anotação de imagens radiológicas usando a web semântica para colaboração científica e clínica*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 3.
- SILVA, L. P. M. *Noções primitivas de Geometria: ponto, reta, plano e espaço*. 2019. Accessed: 29-06-2021. Disponível em: <<https://mundoeducacao.uol.com.br/matematica/noco-es-primitivas-geometria-ponto-reta-plano-espaco.html/>>. Citado 2 vezes nas páginas 5 e 6.
- SOSHNIKOV, D. An approach for creating distributed intelligent systems. In: CITESEER. *Proceedings of the 1st International Workshop on Computer Science and Information Technologies, Moscow, Mephi Publishing*. [S.l.], 1998. p. 129–134. Citado na página 2.
- WATERS, D. *Do microfilme à imagem digital: projeto conservação preventiva em bibliotecas e arquivos*. 2003. Citado na página 4.