



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO - UFERSA
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Efraim Naassom Helem Dantas Rodrigues

Refinamento de Buscas por Imagens Digitais Utilizando Nuvens Dinâmicas

Mossoró-RN

2017

Efraim Naassom Helem Dantas Rodrigues

Refinamento de Buscas por Imagens Digitais Utilizando Nuvens Dinâmicas

Trabalho de Conclusão de Curso apresentado
ao Curso de Ciência da Computação da Uni-
versidade Federal Rural do Semi-Árido como
requisito parcial para a obtenção do grau de
bacharel em Ciência da Computação.

Orientador: Prof. Dr. Leandro Carlos de Souza

Mossoró-RN

2017

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

R685r Rodrigues, Efraim Naassom Helem Dantas.
Refinamento de Buscas por Imagens Digitais
Utilizando Nuvens Dinâmicas / Efraim Naassom Helem
Dantas Rodrigues. - 2017.
51 f. : il.

Orientador: Leandro Carlos de Souza.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência da
Computação, 2017.

1. Nuvens dinâmicas. 2. recuperação de imagens.
3. agrupamento de imagens. 4. internet. I. de
Souza, Leandro Carlos, orient. II. Título.

EFRAIM NAASSOM HELEM DANTAS RODRIGUES

**REFINAMENTO DE BUSCAS POR IMAGENS DIGITAIS UTILIZANDO NUVENS
DINÂMICAS**

Monografia apresentada a Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Ciência da Computação.

Defendida em: 22 / 05 / 2017.

BANCA EXAMINADORA

Leandro Carlos de Souza

Prof. Dr. Leandro Carlos de Souza – UFERSA
Orientador e Presidente

Angélica Félix de Castro

Prof. Dr^a. Angélica Félix de Castro – UFERSA
Primeiro Membro

Flávia Estélia

Prof^a. MSc^r. Flávia Estélia Silva Coelho – UFERSA
Segundo Membro

Soli Deo Gloria

Resumo

Com a popularização dos dispositivos de aquisição de imagens e da *INTERNET*, milhares de imagens estão disponíveis para busca. É possível recuperar e acessar imagens públicas através de motores de busca, como o do *Google*, por exemplo. Em virtude do grande volume de imagens disponíveis em diversos acervos, torna-se necessário o desenvolvimento de métodos que possam selecionar e refinar os resultados desejados. Neste contexto, algoritmos de classificação propõem a organização de elementos de um conjunto utilizando agrupamento e classificação. Assim, é possível utilizar estes algoritmos para organizar resultados de buscas de imagens de tal modo que elas fiquem agrupadas por características afins. Desta forma, este trabalho propõe a criação de um sistema para refinar e organizar conjuntos de imagens resultantes de uma busca na *INTERNET*. Este refinamento consiste em agrupar os resultados semelhantes entre si em pastas, auxiliando o usuário em sua busca. Os métodos aqui estudados envolvem o agrupamento de imagens por meio de nuvens dinâmicas e de descritores que representam as informações contidas nas mesmas.

Palavras-chave: Nuvens dinâmicas. recuperação de imagens. agrupamento de imagens. internet.

Abstract

Due the growing popularity of image acquisition devices and INTERNET, millions of images are available. It is possible to retrieve and access public images through search engines, such as Google's for instance. Because of the huge amount of available images in many repositories, the development of methods that can select and filter desired results is crucial. In this context classification algorithms aim to sort elements of a set using clustering and categorization. Thus it is possible to apply these algorithms to sort results from a search for images in such a manner that they are grouped according to related aspects. Hence, this work aim to create a system to filter and arrange sets of images retrieved from a search on the INTERNET. This filter intent to group similar elements in folders to aid users throughout search. The approaches in this work involve the categorization of images through dynamic clustering and descriptors that represent the information within the images.

Keywords: Dynamic clustering. image retrieval. image clustering. content based image retrieval. internet.

Listas de ilustrações

Figura 1 – (a) Exemplo de imagem digital (b) Representação matricial da imagem.	15
Figura 2 – (a) Imagem contínua (b) Linha de varredura do ponto A a B (c) Amostragem e quantização (d) Linha de varredura digital.	16
Figura 3 – (a) Esquema do cubo de cores RGB (b) Cubo de cores do RGB.	17
Figura 4 – Decomposição de imagem nos canais de cores do sistema RGB.	18
Figura 5 – Imagem colorida ampliada.	19
Figura 6 – Quatro tipos básicos de imagem: escura, clara, baixo contraste, alto contraste e seus histogramas correspondentes.	21
Figura 7 – Organização global de <i>Website</i>	24
Figura 8 – Código 2.1 apresentado no <i>browser</i>	26
Figura 9 – Fluxograma de execução da aplicação.	28
Figura 10 – Caixa de diálogo para entrada do número de grupos desejado.	29
Figura 11 – Resultado de busca.	32
Figura 12 – Exibição do resultado do algoritmo na própria página.	33

Lista de tabelas

Tabela 1 – Imagens de página de teste.	34
Tabela 2 – Matriz de distâncias - Autovetores Dominantes.	35
Tabela 3 – Matriz de distâncias - Histogramas.	35
Tabela 4 – Imagens de página de pesquisa - Chaves.	38
Tabela 5 – Autovetor - Resultado do conjunto chaves em 3 grupos.	39
Tabela 6 – Histograma - Resultado do conjunto chaves em 3 grupos.	40
Tabela 7 – Imagens de página de pesquisa - Lentilha.	41
Tabela 8 – Autovetor - Resultado do conjunto lentilha em 4 grupos.	42
Tabela 9 – Histograma - Resultado do conjunto lentilha em 4 grupos.	43
Tabela 10 – Imagens de página de pesquisa - Casa.	44
Tabela 11 – Autovetor - Resultado do conjunto casa em 5 grupos.	45
Tabela 12 – Histograma - Resultado do conjunto casa em 5 grupos.	46
Tabela 13 – Imagens de página de pesquisa - Gato.	47
Tabela 14 – Autovetor - Resultado do conjunto Gato em 6 grupos.	48
Tabela 15 – Histograma - Resultado do conjunto Gato em 6 grupos.	49

List of Algorithms

1	Método das potências.	20
2	Função <i>Initialize</i> que inicializa os K <i>clusters</i>	22
3	Função <i>Representation</i> que elege os protótipos de cada <i>cluster</i>	22
4	Função <i>Allocation</i> que aloca os elementos.	23
5	Algoritmo de agrupamento por nuvens dinâmicas.	23
6	Aplicação do algoritmo de agrupamento por nuvens dinâmicas em imagens da página de resultados do <i>Google</i>	29
7	Função que verifica se página em questão é página de resultados da busca por imagens.	30

Lista de abreviaturas e siglas

CBIR	Content Based Image Retrieval
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
PDI	Processamento Digital de Imagens
SDK	Software Development Kit
TBIR	Text Based Image Retrieval
URI	Uniform Resource Identifiers
URL	Uniform Resource Locator

Sumário

1	INTRODUÇÃO	12
1.1	Problema	13
1.2	Objetivos	13
1.3	Justificativa	13
1.4	Organização	14
2	REFERENCIAL TEÓRICO	15
2.1	Aquisição de Imagens Digitais	16
2.2	Imagens Coloridas	17
2.3	Descritores de Imagens Digitais	18
2.3.1	Autovetor	19
2.3.2	Histograma	20
2.4	Agrupamento	21
2.4.1	Agrupamento por Nuvens Dinâmicas	22
2.5	Internet	23
2.5.1	HTML e CSS	25
2.5.2	JavaScript	26
2.5.3	Disposição e Recuperação de Informações na Internet	27
3	EXTENSÃO PARA O <i>MOZILLA FIREFOX</i>	28
3.1	Aplicação	28
3.2	Carregamento de Imagens e Geração de Descritores	30
3.3	Agrupamento e Exibição dos Resultados	31
4	RESULTADOS	34
5	CONCLUSÃO E TRABALHOS FUTUROS	50
	REFERÊNCIAS	51

1 Introdução

Os avanços econômicos e tecnológicos das sociedades facilitaram o acesso de grande parte da população a dispositivos tecnológicos com diversas funcionalidades, como permitir a aquisição e publicação de mídias na *INTERNET*. Com uma quantidade massiva de indivíduos em posse destes dispositivos, uma grande quantidade de imagens digitais é gerada e publicada a todo instante na *INTERNET*.

Em função da grande quantidade de mídias disponíveis, a necessidade de organização e sistematização da busca por informações tem sido tratada por grupos e estudiosos de todo o mundo. Diversos serviços de busca, como, por exemplo, *Google*, *Microsoft*, *Yahoo!* e *Ask* foram desenvolvidos com o objetivo de resolver este tipo de problema. Entretanto, no contexto de imagens digitais, a maioria destes serviços suportam buscas relacionadas a termos descritivos das imagens e, em geral, não utilizam características semânticas inerentes às mesmas.

O desenvolvimento de soluções para este tipo de problema se iniciou com base nas técnicas tradicionais de recuperação de imagens, como por exemplo relacionar uma classe de documentos à quantidade de vezes que uma palavra específica é encontrada. Entretanto algumas outras alternativas de métricas têm sido arquitetadas, como criar descritores estatísticos do conteúdo da imagem ou ainda combinar técnicas (LI; DREW; LIU, 2014).

Muitos mecanismos de recuperação de mídias digitais têm proposto soluções que consideram a semântica das mídias. Este tipo de recuperação é conhecido como recuperação de imagens baseada no conteúdo (CBIR) (EAKINS; GRAHAM, 1999). As técnicas CBIR são relativamente novas, mas amplamente usadas para recuperação de imagens em bases de dados de larga escala (JAIN; SINGH, 2011).

O principal objetivo das técnicas de resgate de imagens é auxiliar o usuário em recuperar imagens de maneira eficiente. Entretanto esta tarefa requer que o sistema entenda o real propósito de um usuário em uma busca. Por isso que as recuperações de imagens baseadas em texto (TBIR) e CBIR por si podem ser insatisfatórias. O fato é que a tarefa de entender o objetivo de um usuário em uma busca através de termos ou de uma imagem como parâmetros separados é desafiadora (HARTVEDT, 2010).

Este trabalho propõe uma ferramenta a ser usada no navegador *Mozilla Firefox* que pode auxiliar o usuário na tarefa de busca por imagens. A ferramenta proposta é uma extensão que é ativada quando o navegador carrega uma página de resultados de busca por imagem do *Google*. A extensão não interfere na navegação por outras páginas, mas ao detectar a página de interesse, a aplicação carrega as imagens que estão sendo exibidas no *browser* e as separa em grupos definidos por usuário de acordo com as semelhanças encontradas nos conteúdos das imagens.

1.1 Problema

Algoritmos para filtragem e organização de mídias associadas a textos são conhecidos e aplicados a serviços de busca. No contexto de busca de imagens por palavra-chave, as ferramentas tomam como parâmetro os termos associados à imagem desconsiderando o conteúdo da imagem. O problema é que estes filtros ou organização proposta por estas ferramentas podem não corresponder ao nível de semelhança esperado pelo usuário.

Os serviços de buscas de imagens *on-line* não incluem, em seus algoritmos de filtragem de conteúdo, mecanismos de ordenação que fazem referências ao conteúdo das imagens. Desta forma as requisições retornam um grande número de imagens que não são organizadas levando em conta seus conteúdos. Assim, cabe ao usuário final a tarefa de filtrar visualmente o conteúdo das imagens a fim de se identificar quais delas correspondem ao resultado esperado.

1.2 Objetivos

O objetivo deste trabalho é a implementação de um algoritmo capaz de classificar e organizar conjuntos de imagens resultantes do motor de busca do *Google* a partir de semelhanças que as imagens apresentem em seus conteúdos. Para isto, pretende-se aplicar o método de nuvens dinâmicas, método de agrupamento indicado para dados complexos, nos autovetores dominantes e histogramas das imagens.

Para alcançar os objetivos gerais, este trabalho tem como objetivos específicos:

- Ter acesso a imagens de uma página de resultados do serviço de busca do *Google*;
- Realizar a análise do conteúdo de imagens de uma página *Web* e gerar descritores;
- Empregar o algoritmo de agrupamento por nuvens dinâmicas;
- Manipular a página *Web* para exibir as imagens organizadas em grupos na própria página.

1.3 Justificativa

Áreas como medicina, meteorologia e astronomia baseiam suas atividades em buscas feitas em volumosas bases de imagens que tendem a continuar crescendo. A busca por padrões e extração de conhecimento de base de dados tem se mostrado uma atividade vantajosa, não só para a ciência mas também para a economia (BERRY; LINOFF, 1997).

Álbuns digitais (e.g., *Flickr*, *Photobucket*, *Picasa*) e acervos de imagens de propósito geral são apontados como responsáveis para a tendência no aumento de publicação de imagens na *INTERNET* (HARTVEDT, 2010). Entretanto, estes serviços não dispõem de mecanismos

de organização que levam em conta o conteúdo das imagens. Então, busca-se aperfeiçoar a experiência de usuários no uso de ferramentas de resgate de imagens.

1.4 Organização

Além deste capítulo introdutório, este documento é constituído de mais quatro capítulos.

Capítulo 2 - Referencial Teórico

Este capítulo introduz os conceitos utilizados no desenvolvimento da aplicação proposta. A definição de imagem é apresentada, bem como a representação digital e os conceitos envolvidos no processo de sua aquisição. Além disso, são apresentados os conceitos de descritores e do algoritmo de agrupamento por nuvens dinâmicas. Também são expostos os conceitos essenciais implicados no funcionamento da *INTERNET*.

Capítulo 3 - Extensão para o Mozilla Firefox

O capítulo 3 descreve a metodologia utilizada para a solução do problema. As etapas de execução da aplicação são apresentadas.

Capítulo 4 - Resultados

O capítulo 4 apresenta uma discussão acerca dos resultados obtidos.

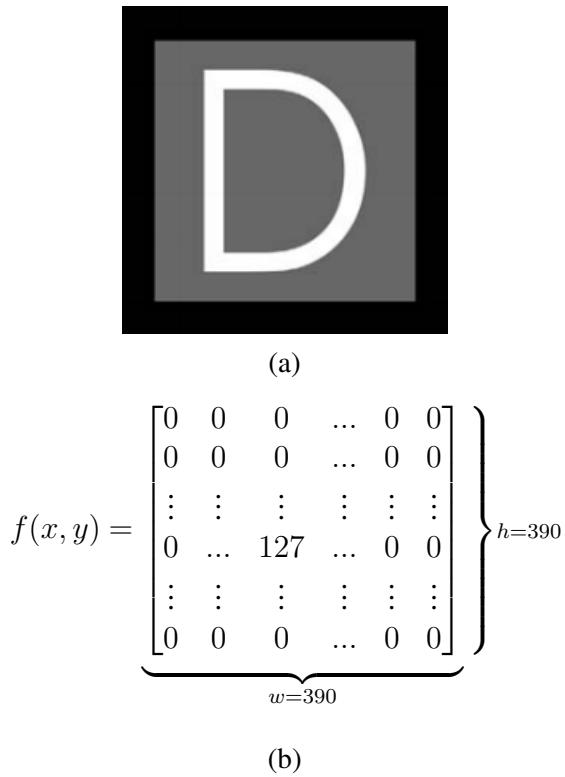
Capítulo 5 - Conclusão e Trabalhos Futuros

Este capítulo expõe as percepções da solução proposta e direcionamentos que podem aprimorar a qualidade dos resultados e eficiência da aplicação.

2 Referencial Teórico

Uma imagem pode ser definida por uma função bidimensional $f(x, y)$ onde x e y são um par de coordenadas espaciais e f em qualquer par de coordenadas (x, y) representa a intensidade da imagem naquele ponto. Quando x, y e os valores das intensidades de f são todos finitos e em quantidades discretas, pode-se dizer que a imagem é uma imagem digital. Os elementos constituintes das imagens são denominados *pels* ou *pixels* (GONZALEZ; WOODS, 2008). A quantidade de *pixels* nas dimensões de altura e largura indicam a resolução de uma imagem. A Figura 1(a), mostra um exemplo de imagem monocromática digital cuja resolução é de 390x390, é representada matricialmente na Figura 1(b).

Figura 1 – (a) Exemplo de imagem digital (b) Representação matricial da imagem.



Fonte: Gonzalez e Woods (2008).

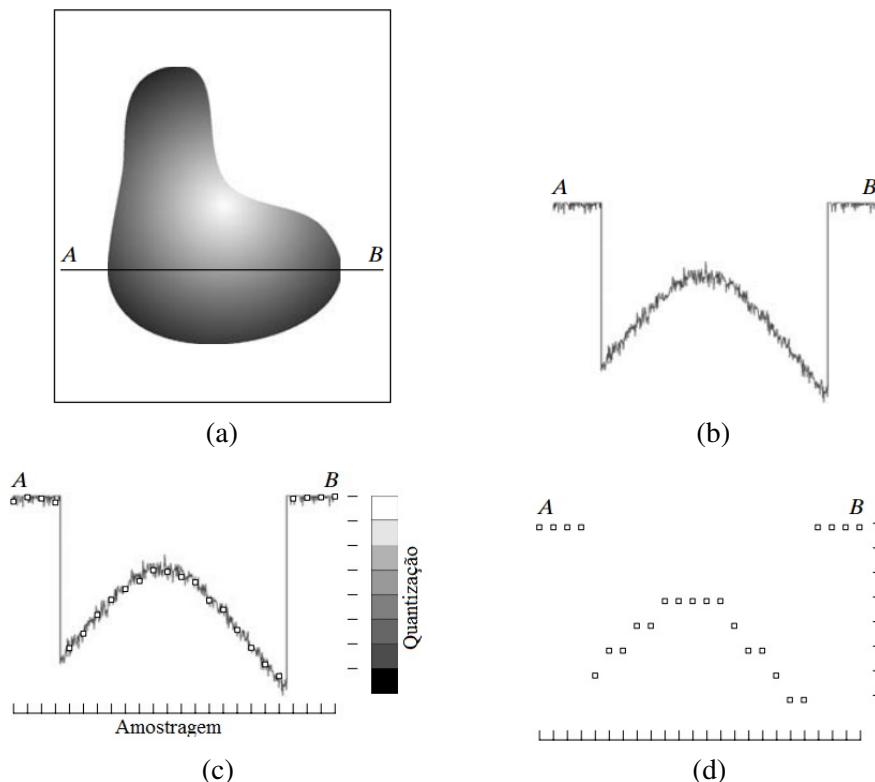
O processamento digital de imagens (PDI) refere-se à utilização de métodos que manipulam imagens digitais e que geram como resultado outras imagens digitais. Apesar de a visão humana ser limitada ao espectro eletromagnético, dispositivos digitais podem operar em quase todo o espectro eletromagnético, realizando assim, operações com imagens geradas de origens não comuns aos humanos, como ultrassom e microscópio eletrônico. Sendo assim, a área de PDI engloba uma grande variedade de campos de aplicações (GONZALEZ; WOODS, 2008).

2.1 Aquisição de Imagens Digitais

Em geral, a aquisição de imagens digitais é feita através de equipamentos digitais de captura de imagens como câmeras, por exemplo. Os dispositivos de captura possuem componentes sensíveis às ondas eletromagnéticas refletidas por objetos presentes em uma cena. Desta forma, uma imagem digital é gerada pela conversão de dados analógicos em dados digitais. O processo executado por estes equipamentos é dividido em três etapas: amostragem, quantização e codificação.

A amostragem é o processo responsável por converter coordenadas contínuas em discretas (BOVIK, 2005). A quantização é a operação responsável por converter as intensidades de luz de cada ponto da imagem em uma representação numérica discreta (WIDROW, 1956). Uma vez que as etapas de amostragem e quantização são passadas, se faz necessário que os valores discretos obtidos na fase de quantização sejam codificados em sequências de bits para que assim possam ser considerados digitais. A Figura 2 apresenta graficamente as etapas envolvidas na aquisição de imagens digitais.

Figura 2 – (a) Imagem contínua (b) Linha de varredura do ponto A a B (c) Amostragem e quantização (d) Linha de varredura digital.



Fonte: Gonzalez e Woods (2008).

2.2 Imagens Coloridas

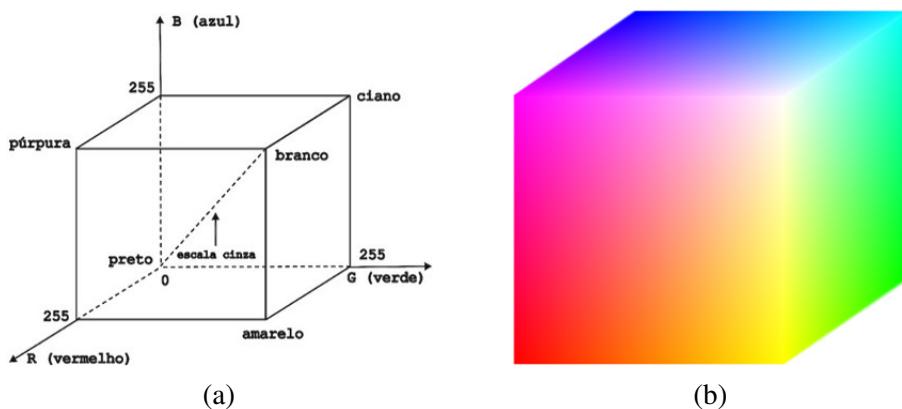
Estudos mostram que os sensores do olho humano podem ser divididos em três principais categorias que correspondem basicamente as cores vermelha, verde e azul (LI; DREW; LIU, 2014). Sendo assim, a literatura considera estas cores como sendo cores primárias. Isto significa que combinações destas cores geram outras cores (JAIN, 1989).

Como a área de PDI engloba processos que têm como entrada e saída imagens, diferentes sistemas de cores foram estudados para dar suporte ao processo de exibição de imagens de acordo com a limitação e especificidade de cada dispositivo de saída de imagens. São exemplos de dispositivos de saídas que necessitam de diferentes sistemas de cores a impressora, televisão colorida e dispositivos de plasma.

Em termos de PDI, os modelos de cores mais comuns são RGB (*red, green, blue*) para monitores coloridos e uma grande parte de câmeras de vídeo colorido, CMY (*cyan, magenta, yellow*) e CMYK (*cyan, magenta, yellow, black*) para impressão colorida, e HSI (*hue, saturation, intensity*) (GONZALEZ; WOODS, 2008). O RGB representa o sistema de cores que usa as cores vermelho, verde e azul como cores primárias, o CMY as cores ciano, magenta e amarelo, o CMYK as cores do CMY com adição da cor preta, por fim o HSI que refere-se à tonalidade, saturação e intensidade.

No modelo RGB, cada cor aparece em seus componentes primários, vermelho, verde, e azul. Este modelo é baseado em um sistema de coordenadas cartesianas. O subespaço de cores é representado no cubo da Figura 3(a). Neste modelo, a escala de cinza se estende desde o preto, na origem até o branco no canto mais distante da origem (LI; DREW; LIU, 2014).

Figura 3 – (a) Esquema do cubo de cores RGB (b) Cubo de cores do RGB.



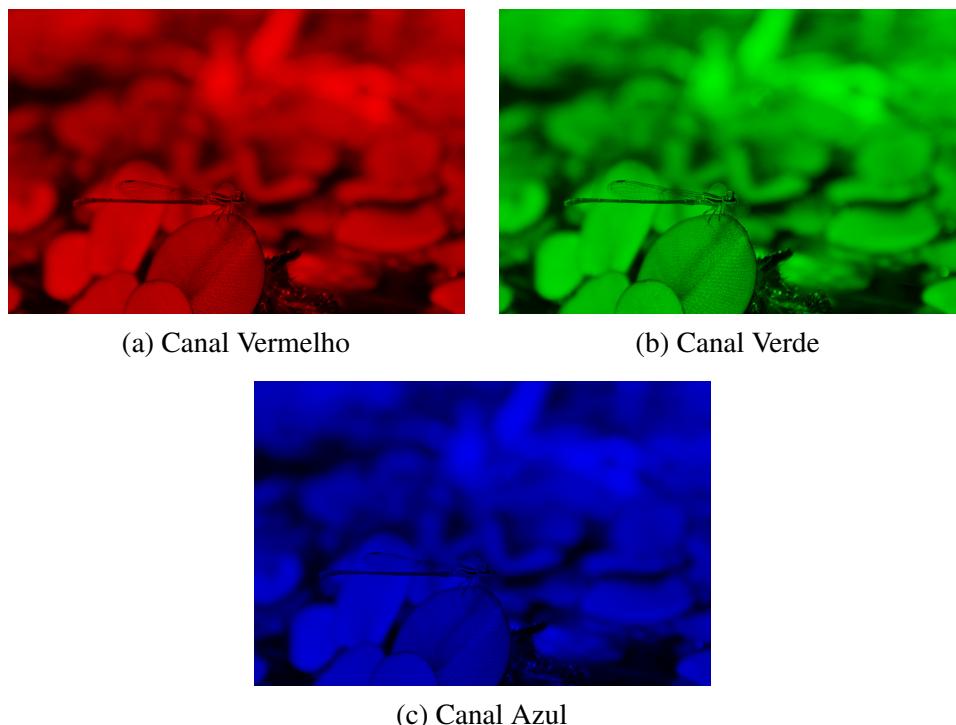
Fonte: Gonzalez e Woods (2008).

Como visto anteriormente, uma imagem monocromática é definida por uma função bidimensional $f(x, y)$ e pode ser representada computacionalmente por uma matriz bidimensional de forma que cada célula da matriz representa uma intensidade de cor da imagem. Sendo assim, uma

imagem representada no sistema RGB possui três matrizes de forma que cada matriz representa as intensidades de cada cor deste sistema. Neste modelo, cada célula da matriz possui um valor que é representado por 8 bits que com a combinação de três cores permite a composição de $(2^8)^3 = 16.777.216$ cores.

As Figuras 4(a), 4(b) e 4(c) representam a decomposição de uma imagem colorida nos três canais de cores do sistema RGB. Cada imagem representa a intensidade de uma respectiva cor na imagem. Sendo assim, a imagem colorida é formada pela combinação destas três imagens. A Figura 5 é um exemplo de imagem colorida ampliada. Cada *pixel* é descrito por uma combinação das três intensidades de cores.

Figura 4 – Decomposição de imagem nos canais de cores do sistema RGB.



Fonte: Autoria Própria.

2.3 Descritores de Imagens Digitais

Um descritor é uma representação resumida para a informação contida no interior da imagem. Dependendo do descritor utilizado, algumas características podem ser preservadas e outras perdidas. Na literatura existem e são propostos vários descritores de imagens: descritores de cores, textura e forma (TUYTELAARS; MIKOLAJCZYK et al., 2008; GONZALEZ; WOODS, 2008). Neste trabalho foram utilizados dois descritores, autovetor e histograma, que serão utilizados como critério de comparação entre as imagens. Estes descritores foram adotados por

serem suficientes para representar as imagens sem que precisem ser complementados por outros descriptores.

Figura 5 – Imagem colorida ampliada.

107	98	100	105	131	113	119	143	171	131	171	1
18	9	11	18	46	33	42	68	108	155	1	
62	55	57	61	88	71	78	101	135	165	1	
97	85	76	77	131	142	120	137	168	205	2	
16	13	8	14	63	61	35	58	100	158	1	
59	61	59	59	104	94	66	89	123	166	1	
99	88	79	83	102	127	118	99	114	199	2	
21	13	6	16	34	56	43	24	48	148	1	
60	56	51	57	71	86	74	57	76	163	1	
122	97	94	87	110	156	128	99	111	191	2	
46	15	14	16	43	87	57	27	45	142	1	
82	53	53	48	74	116	91	65	81	164	2	
140	127	126	115	139	162	119	98	127	202	2	
60	43	42	41	66	88	44	27	65	160	1	
95	76	75	68	95	121	83	71	106	184	2	
152	136	135	138	133	124	110	105	159	218	2	
68	56	55	62	54	42	32	38	107	185	2	
102	85	82	90	86	80	74	82	143	204	2	
135	141	144	137	127	107	116	145	200	226	2	
49	60	63	60	47	26	40	89	160	198	1	
86	92	95	92	86	69	86	128	187	210	2	
111	108	115	100	112	122	137	186	220	229	2	
26	24	30	24	39	53	79	145	189	199	1	
65	60	69	63	82	98	120	175	207	207	1	
134	122	123	121	128	162	186	208	220	222	2	
54	34	34	44	60	107	144	179	194	190	1	
93	76	78	90	107	149	180	201	207	195	1	
162	150	144	153	169	192	206	220	219	224	2	
76	71	73	91	117	148	170	189	187	187	1	
123	126	129	142	156	171	182	195	192	194	1	
187	182	184	186	196	214	226	231	230	234	2	

Fonte: Montabone (2007).

2.3.1 Autovetor

Um vetor é uma grandeza com módulo, direção e sentido. Os vetores são representados por retas que ligam dois pontos. O ponto que define a ponta da reta é chamado de ponto final e o outro ponto é chamado de ponto de origem.

O autovetor de uma matriz quadrada $A_{n \times n}$ é um vetor v que atende a Equação 2.1 onde $\lambda \in \mathbb{R}$ é o autovalor de A (BOLDRINI et al., 1986). Quando $A_{n \times n}$ é uma imagem, o autovetor dominante mantém os aspectos mais efetivos da imagem de forma que v indica a direção de maior variação da informação, neste caso a informação de cor.

$$Av = \lambda v \quad (2.1)$$

Os autovetores são vetores representados por um ponto de origem na origem do espaço e um ponto final que é calculado através do método das potências. O método das potências é um método iterativo aplicado a uma matriz quadrada $A_{n \times n}$ que produz uma sequência de vetores v_i que convergem ao autovetor dominante. O método das potências é apresentado por Saad (2011) e Zhang (2009) e é exibido no Algoritmo 1.

Algoritmo 1 Método das potências.

-
- 1: Seja v_0 um vetor não nulo.
 - 2: **para** $i \leftarrow 0, 1, 2, \dots$ até convergir **faca**
 - 3: $v_{i+1} \leftarrow Av_i$
 - 4: $v_{i+1} \leftarrow \frac{v_{i+1}}{\|v_{i+1}\|}$
 - 5: **fim para**
-

A distância calculada entre duas imagens representadas por autovetores v e w é feita com base no ângulo entre estes vetores. Boldrini et al. (1986) mostra que o ângulo entre dois vetores é calculado através da Equação 2.2.

$$\theta = \arccos\left(\frac{\langle v, w \rangle}{\|v\|\|w\|}\right) \quad (2.2)$$

2.3.2 Histograma

O histograma de uma imagem digital $A_{n \times m}$ é a função discreta $h(r_k) = n_k$, de forma que r_k são os valores de intensidades da imagem e n_k é o número de *pixels* na imagem com intensidade r_k . O histograma pode ser normalizado através da divisão de cada componente pelo número total de *pixels* da imagem $n \times m$. Sendo assim, um histograma normalizado é denotado por $p(r_k) = n_k/nm$ de forma que $p(r_k)$ representa a probabilidade de ocorrência da intensidade r_k (GONZALEZ; WOODS, 2008).

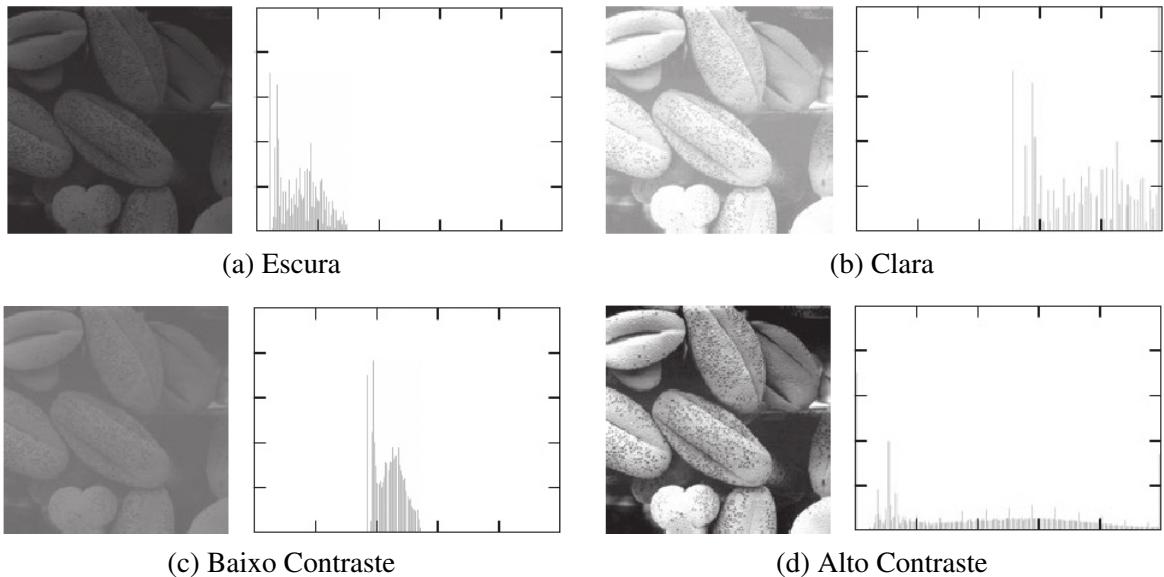
A visualização gráfica de um histograma se dá através da relação entre $h(r_k) = n_k$ e r_k ou $p(r_k) = n_k/nm$ e r_k como apresenta a Figura 6. A representação gráfica de um histograma pode ser utilizada para analisar características visuais da imagem e possivelmente identificar particularidades na imagem, como identificar as cores predominantes ou o nível de contraste.

Na Figura 6(a) os componentes do histograma estão concentrados no lado esquerdo da escala de intensidade que representa os baixos níveis de intensidade. Semelhantemente, os componentes do histograma da Figura 6(b) tendem à direção do lado direito da escala. A Figura 6(c) com baixo contraste tem um histograma graficamente estreito que concentra as ocorrências das intensidades ao centro. Já na Figura 6(d), os componentes do histograma são distribuídos de maneira mais uniforme.

A distância entre dois histogramas x e y é calculada utilizando a distância euclidiana que é usada para calcular distâncias no \mathbb{R}^n . A distância euclidiana é apresentada na Equação 2.3 (DEZA; DEZA, 2009).

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (2.3)$$

Figura 6 – Quatro tipos básicos de imagem: escura, clara, baixo contraste, alto contraste e seus histogramas correspondentes.



Fonte: Gonzalez e Woods (2008).

2.4 Agrupamento

Agrupamento é o nome dado ao processo de agrupar ou categorizar elementos. Este processo tem como entrada um conjunto de elementos sem organização prévia e busca identificar possíveis grupos de elementos que se assemelham entre si. Sendo assim, elementos semelhantes tendem a ser alocados nos mesmos grupos e consequentemente elementos distintos tendem a ficar em grupos distintos (CARVALHO; LECHEVALLIER; de Souza, 2004).

Os grupos identificados neste processo também podem ser chamados de *clusters*. O processo de identificação dos *clusters* pode ser chamado de clusterização.

A clusterização de um conjunto de elementos pode acontecer de maneira supervisionada e não-supervisionada. O processo de clusterização supervisionado requer a intervenção humana enquanto que a clusterização não supervisionada não requer a intervenção humana. O algoritmo de agrupamento por nuvens dinâmicas usado neste trabalho se encaixa no grupo de algoritmo de clusterização não supervisionado.

As estratégias de clusterização podem ser divididas em dois tipos principais: hierárquicos e particionais. Os algoritmos hierárquicos são divididos em aglomerativos e divisivos. Os algoritmos hierárquicos organizam os conjuntos de dados em estruturas hierárquicas denominadas de dendogramas e com isso aplicam sucessivas divisões dos elementos (CARVALHO, 2016).

As estratégias de agrupamentos particionais buscam dividir um conjunto de elementos de entrada em um número fixo de *clusters*. Estas estratégias utilizam pontos que se localizam

no mesmo espaço dimensional dos elementos para representar cada um dos grupos formados (CARVALHO, 2016). Estes pontos são denominados de protótipos e são usados como parâmetro de distância de um elemento ao *cluster*.

2.4.1 Agrupamento por Nuvens Dinâmicas

O algoritmo de agrupamento por nuvens dinâmicas é um algoritmo iterativo de agrupamento particional que itera sobre passos de alocação de elementos e representação de *clusters* até que o critério de parada seja alcançado.

O problema de otimização resolvido por este algoritmo é definido por Carvalho, Lechevallier e de Souza (2004) como se segue. Seja Ω o conjunto de n objetos indexados por $i = 1, \dots, n$ e descritos por p variáveis quantitativas. Cada objeto i é descrito por um vetor $x_i = (x_i^1, \dots, x_i^p) \in \mathbb{R}^p$. O problema é achar a partição $P = (C_1, \dots, C_k)$ de Ω em K *clusters* e o conjunto $Y = (y_1, \dots, y_K)$ de exemplares que minimizam o critério de particionamento $g(P, Y)$ que mede a adequação entre os *clusters* e seus representantes.

A inicialização do algoritmo se dá pela declaração inicial das partições e de seus respectivos protótipos como mostra o Algoritmo 2. Uma vez inicializado, o algoritmo busca encontrar a melhor alocação para os elementos de maneira que o critério de particionamento $g(P, Y)$ seja minimizado. Para tal, os elementos são alocados nos *clusters* de acordo com suas proximidades aos exemplares (CARVALHO; LECHEVALLIER; de Souza, 2004).

Algoritmo 2 Função *Initialize* que inicializa os K *clusters*.

- 1: **função** INITIALIZE
 - 2: Escolha uma partição (C_1, \dots, C_K) de Ω ou escolha K objetos diferentes y_1, \dots, y_K entre Ω e aloque cada objeto i ao exemplar mais próximo para construir a partição inicial (C_1, \dots, C_K) .
 - 3: **fim função**
-

O Algoritmo 3 mostra que a eleição dos protótipos acontece através da verificação de qual elemento está mais próximo ao restante dos elementos, este elemento representará o centro do *cluster* e por isso pode ser chamado de centroide. Em algumas configurações do algoritmo, o representante do *cluster* pode ser calculado como a média de todos os elementos presentes naquele *cluster*. O centroide de um *cluster* poderá ser usado como parâmetro para o cálculo de distância de um elemento ao *cluster*.

Algoritmo 3 Função *Representation* que elege os protótipos de cada *cluster*.

- 1: **função** REPRESENTATION
 - 2: **para** $k \leftarrow 0; i < K; k + +$ **faça**
 - 3: Calcule o exemplar y_k
 - 4: **fim para**
 - 5: **fim função**
-

O Algoritmo 4 apresenta a função responsável por alocar os elementos em seus respectivos *clusters*. Esta etapa do algoritmo de agrupamento por nuvens dinâmicas percorre todos os elementos averiguando o *cluster* em que o elemento melhor se encaixa (menor distância) para que a melhor configuração seja estabelecida.

Algoritmo 4 Função *Allocation* que aloca os elementos.

```

1: função ALLOCATION
2:   teste  $\leftarrow 0$ 
3:   para  $i \leftarrow 0; i < n; i++$  faca
4:     seja o cluster  $C_{k*}$  tal que  $k* = \min_{k=0,\dots,K} d_r^k(x_i, y_k)$ 
5:     se  $i \in C_k \quad E \quad k* \neq k$  então
6:       teste  $\leftarrow 1$ 
7:        $C_{k*} \leftarrow C_{k*} \cup \{i\}$ 
8:        $C_k \leftarrow C_k \setminus \{i\}$ 
9:     fim se
10:   fim para
11:   retorne teste
12: fim função
  
```

A linha 2 do Algoritmo 4 declara a variável que indica quando uma mudança na configuração dos *clusters* for efetivada. Esta mudança é essencial para o algoritmo de agrupamento por nuvens dinâmicas, uma vez que esta indicará a necessidade de mais uma iteração do algoritmo.

O Algoritmo 5 reúne as etapas apresentadas para contemplar a proposta do algoritmo de agrupamento. Uma vez que os *clusters* são inicializados, o algoritmo itera pelas etapas de representação (REPRESENTATION) e alocação (ALLOCATION) até que nenhuma melhoria possa ser feita pela etapa de alocação.

Algoritmo 5 Algoritmo de agrupamento por nuvens dinâmicas.

```

1: função DYNAMICCLUSTERING
2:   INITIALIZE
3:   repita
4:     REPRESENTATION
5:     teste  $\leftarrow$  ALLOCATION
6:   até teste  $= 0$ 
7: fim função
  
```

2.5 Internet

O avanço e estruturação das redes de computadores possibilitaram que várias aplicações que se baseiam na comunicação em redes surgissem. A *INTERNET* refere-se a arquitetura de redes globais que estão interconectadas e se comunicam através de conjuntos de regras e protocolos. A *INTERNET* usa funcionalidades relativamente simples com escalabilidade, eficiência e utilidade suficientes para resultarem em um espaço confiável (W3C, 2004).

A *World Wide Web* (WWW) ou simplesmente *Web* é uma das diversas aplicações de escala mundial que possibilitam que instituições e pessoas de todo o mundo estejam conectadas através da *INTERNET*.

A *World Wide Web* pode ser considerada um enorme sistema para acessar documentos ligados, que consiste em milhões de clientes e servidores. Servidores mantêm conjuntos de documentos, enquanto clientes fornecem a usuários uma interface de fácil utilização para apresentar e acessar esses documentos (TANENBAUM; STEEN, 2007).

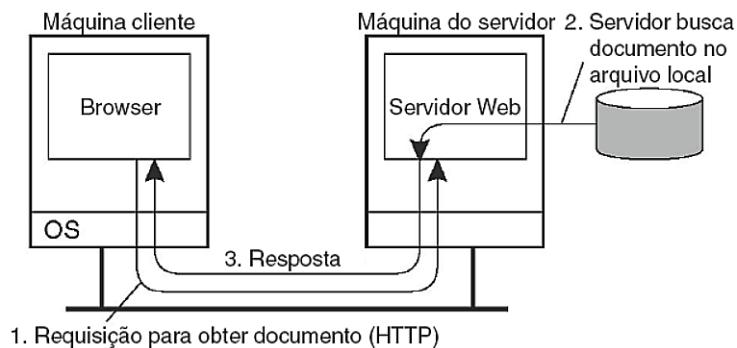
O padrão *Web* teve início no Laboratório Europeu de Física Nuclear (*European Particle Physics Laboratory* - CERN), em Genebra, como um projeto que permitiria seu grupo de pesquisadores, numeroso e geograficamente disperso, acessar documentos compartilhados por meio de um sistema simples de hipertexto. Um documento podia ser qualquer coisa que pudesse ser representada no terminal do computador de um usuário (TANENBAUM; STEEN, 2007).

O núcleo de um site *Web* é formado por um processo que tem acesso a um sistema de arquivos local que armazena documentos. O modo mais simples de referenciar um documento é por meio de uma referência denominada localizador uniforme de recurso ou *Uniform Resource Locator* (URL) (TANENBAUM; STEEN, 2007). Cada documento disponível na *Web* possui um URL associado que será usado para que o serviço cliente tenha acesso a estes documentos.

A comunicação entre um *browser* e um servidor se baseia no protocolo de transferência de hipertexto ou *Hypertext Transfer Protocol* (HTTP). O protocolo HTTP foi originalmente projetado para transferência de conteúdo *Web*, mas agora suporta a transferência de arquivos de qualquer tipo (LI; DREW; LIU, 2014).

O *browser* ou navegador é a aplicação responsável por abstrair os procedimentos envolvidos na requisição e tratamento da resposta obtida. A Figura 7 ilustra o procedimento em que um cliente realiza uma requisição HTTP e obtém uma resposta do servidor. Uma vez que a máquina cliente recebe uma resposta do servidor com o conteúdo esperado, o *browser* analisa a resposta e apresenta o conteúdo do documento.

Figura 7 – Organização global de Website.



Fonte: Tanenbaum e Steen (2007).

2.5.1 HTML e CSS

Uma página *Web* é formada principalmente de código HTML, uma linguagem de marcação usada para indicar ao *browser* como estruturar as páginas. HTML consiste em uma série de elementos usados para delimitar ou marcar diferentes partes do conteúdo para que sejam exibidos de maneira específica de acordo com o elemento indicado (Mozilla Developer Network, 2017). Os elementos são delimitados por *tags* que definem o início de um elemento no formato `<tag>` e `</tag>` que delimita o fim do elemento.

Um documento HTML simples é dividido entre cabeçalho e corpo. O cabeçalho descreve as definições do documento que são interpretadas antes do processo de montagem do documento ser concluído. Estas definições incluem título da página, *links* para recursos utilizados e outras informações que o autor desejar especificar. O corpo de uma página descreve a estrutura do documento e seu conteúdo. Estas estruturas podem ser parágrafos, tabelas, formulários, listas de itens e botões (LI; DREW; LIU, 2014).

O Código 2.1 mostra o exemplo de uma página simples HTML. As linhas 1 e 10 delimitam que o texto em seu interior é um código HTML, as 2 e 4 delimitam o cabeçalho da página, as 6 e 9 delimitam o corpo. No interior destes elementos, a linha 3 declara o título da página, a linha 7 declara o título do texto e a linha 8 declara um parágrafo. O resultado deste exemplo em um *browser* é apresentado na Figura 8.

```
1 <html>
2 <head>
3   <title>Lorem ipsum dolor sit amet</title>
4 </head>
5
6 <body>
7   <h1>Lorem ipsum dolor sit amet</h1>
8   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed nulla enim. Ut dolor
      nisi, consectetur sed iaculis non, sollicitudin id magna.</p>
9 </body>
10 </html>
```

Código 2.1 – Exemplo de página HTML

Os elementos HTML podem conter atributos que adicionam informações extras sobre o elemento. Um atributo permite que um elemento específico passe a ter uma identificação que pode ser usada para estilizar este elemento. A estilização de uma página e de seus elementos é feita através da linguagem *Cascading Style Sheets* (CSS). A linguagem CSS permite a customização de cores de texto e fundo dos elementos, fontes, tamanhos e espaçamentos entre outros (Mozilla Developer Network, 2017).

Figura 8 – Código 2.1 apresentado no *browser*.

Lorem ipsum dolor sit amet

Consectetur sed iaculis non, sollicitudin id magna.

Fonte: Autoria Própria.

2.5.2 JavaScript

Os *browsers* não se limitam apenas a apresentar as páginas Web estáticas, eles permitem que os usuários interajam com páginas através da linguagem *JavaScript*. Apesar do nome, *JavaScript* e *Java* são linguagens totalmente diferentes. A linguagem *JavaScript* é uma linguagem de *script* que é interpretada por um *browser*, isto significa que os *scripts JavaScript* são executados por a máquina cliente sem a necessidade de passar por um compilador.¹

JavaScript é uma linguagem de programação que permite a implementação de interações complexas nas páginas *Web*. Ela permite a atualização de componentes HTML e regras CSS de maneira dinâmica e permite armazenar valores em variáveis, realizar operações em textos e executar ações em resposta a eventos que venham a acontecer na página (Mozilla Developer Network, 2017).

O Código 2.2 é um exemplo de como *JavaScript* pode interagir com HTML e CSS. Neste exemplo, o botão definido na linha 5 define o código *JavaScript* a ser executado quando um evento de clique for detectado no botão. A ação declarada no atributo *onclick* do botão HTML (`document.getElementById('paragrafo').style.display = 'none'`) acessa o elemento HTML da linha 4 através do identificador definido no atributo *id* e altera o atributo CSS *display* para *none* fazendo com que este fique invisível.

```
1 <html>
2 <body>
3 <h1>Lorem ipsum</h1>
4 <p id="paragrafo">Isto vai sumir</p>
5 <button type="button" onclick="document.getElementById('paragrafo').style.display='none'">
   /button>
6 </body>
7 </html>
```

Código 2.2 – Exemplo de página HTML com código *JavaScript* interagindo com CSS

Os códigos *JavaScript* são executados por um processador de código *JavaScript* contido no *browser* depois que os códigos *HTML* e *CSS* são carregados. Isto porque os *scripts JavaScript*

¹ Script é uma sequência de instruções a ser interpretada por um programa em vez de ser compilada.

interagem com os componentes da página e precisam que eles estejam carregados.

2.5.3 Disposição e Recuperação de Informações na Internet

É fato que a *Web* é um serviço democrático e dinâmico assim como o mundo virtual em geral. Com a popularização da *INTERNET* e dos dispositivos digitais, artefatos multimídias dos mais diversos formatos são gerados a cada instante. Com isso, é comum que estas mídias sejam disponibilizadas na *WWW* em *Websites* dos mais diversos tipos.

Com o alto volume de dados sendo disponibilizados na *INTERNET* nos mais variados formatos, a tarefa de encontrar ou recuperar uma informação de maneira não metódica é inviável pois os documentos disponibilizados não seguem qualquer ordem. É por isso que vários serviços como *Google*, *Yahoo* e *Bing* se dispõem a realizar buscas na *INTERNET* de maneira eficiente. O Net Market Share (2017) indica que o motor de busca do *Google* dominou cerca de 73% deste mercado.

O *Google* é uma corporação dos Estados Unidos cuja missão é dominar o enorme volume de informações: “*organizar as informações do mundo e torná-las universalmente acessíveis e úteis*” (Google, 2017).

Inaugurada em 1998, o *Google* cresceu até ter uma fatia dominante do mercado de pesquisa na *INTERNET* principalmente devido à eficiência de seu algoritmo. Desde seu sistema de produção inicial, até lidar com mais de 88 bilhões de pesquisas por mês (em 2010), seu principal mecanismo de busca nunca sofreu uma interrupção em todo esse tempo e seus usuários podem esperar resultados de consulta em torno de 0,2 segundos (COULOURIS et al., 2013).

A seguir é apresentada uma síntese de uma breve apresentação do serviço de busca do *Google* feita por Coulouris et al. (2013). A função do mecanismo de busca do Google, assim como para qualquer mecanismo de pesquisa na *Web*, é receber determinada consulta e retornar uma lista ordenada com os resultados mais relevantes que satisfazem essa consulta, pesquisando o conteúdo na *Web*. O mecanismo de busca do *Google* consiste em um conjunto de serviços para esquadrinhar a *Web* e indexar e classificar as páginas descobertas.

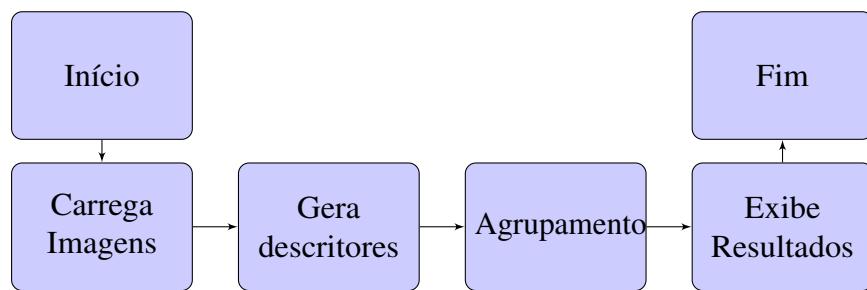
A tarefa de esquadrinhar refere-se a uma análise minuciosa do conteúdo da *Web*. Esta tarefa analisa as páginas recursivamente coletando todos os *links* que apontam para outras páginas para que estas também sejam esquadrinhados por este procedimento. O objetivo é mapear todo o conteúdo disponível na *INTERNET* para que seja gerado um índice de palavras ou termos que serão usados como parâmetros de busca.

O índice de palavras gerado adota critérios de comparação e ordenação como número de ocorrências, posição no documento e tamanho da fonte usada. Além do índice de palavras, o motor de busca mantém um índice de *links* que monitora as páginas que ligam a um determinado *Website*. Este índice de *links* é usado por o algoritmo *PageRank* que classifica as páginas de acordo com sua relevância.

3 Extensão para o Mozilla Firefox

Este capítulo trata da aplicação do algoritmo de agrupamento por nuvens dinâmicas para o refinamento de buscas por imagens na *INTERNET* usando o motor de busca do *Google*. O serviço do *Google* de pesquisa oferece por si uma organização dos resultados em grupos. A diferença da organização do *Google* para a organização proposta é que enquanto os resultados e a busca têm como parâmetro de diferenciação o texto associado aos componentes de busca, este trabalho propõe que a disposição dos resultados seja feita com base no conteúdo das imagens. O fluxo de passos executados pela extensão é representado na Figura 9.

Figura 9 – Fluxograma de execução da aplicação.



Fonte: Autoria Própria.

3.1 Aplicação

A extensão do *Mozilla Firefox* foi desenvolvida em *JavaScript* e não altera o modo com que o usuário navega na *INTERNET*, então para que a execução do código seja iniciada é necessário que o usuário acesse a página de resultados de pesquisas de imagens do *Google*. O *Mozilla Firefox* foi escolhido por se tratar de um navegador de código aberto desenvolvido sem fins lucrativos, além disso, o *Mozilla Firefox* é adotado como navegador padrão na maioria dos sistemas operacionais de código aberto e de licença livre.

O Algoritmo 6 é um esboço do que é feito pela aplicação proposta. A extensão verifica a página que está sendo acessada a cada página carregada no navegador, quando a página em questão é a página específica, a extensão exibe uma caixa de diálogo *JavaScript* para que o usuário informe o número de *clusters* desejados como mostra a Figura 10. De maneira semelhante, o usuário escolhe o descriptor a ser usado pelo algoritmo. Isto feito, as imagens são carregadas, os descritores são gerados, o algoritmo executado e o resultado apresentado na própria página.

As imagens que são mostradas como resultado de uma busca por imagens na página do *Google* são os parâmetros de entrada do código. Assim como a maioria dos *Websites* na

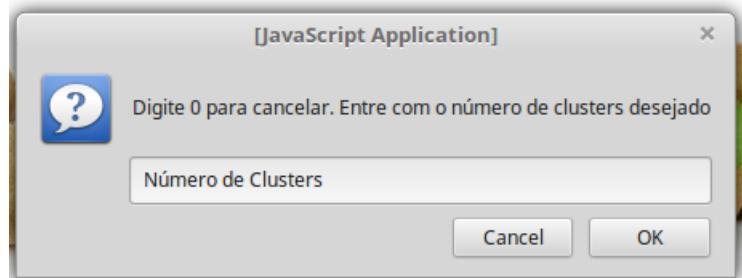
INTERNET as páginas de resultados de buscas do *Google* são compostas por componentes HTML, CSS e códigos *JavaScript*.

Algoritmo 6 Aplicação do algoritmo de agrupamento por nuvens dinâmicas em imagens da página de resultados do *Google*.

```

1: função AGRUPARIMAGENS(document)
2:   se ISRESULTPAGE(document) então
3:     imagens ← document.getElementsByClassName("rg_ic rg_i")
4:     descritores ← GERARDESCRITORES(imagens)
5:     clusters ← DYNAMICCLUSTERING(descritores)
6:     SHOWRESULTS(clusters)
7:   fim se
8: fim função
```

Figura 10 – Caixa de diálogo para entrada do número de grupos desejado.



Fonte: Autoria Própria.

A obtenção das imagens da página de resultados do *Google* é possível através de *scripts JavaScript* que são capazes de acessar os elementos HTML e suas propriedades. Como descrito anteriormente, os códigos HTML são organizados por componentes e divisores e estes são úteis para exibir o conteúdo da página de maneira estilizada para cada componente e divisão.

O Código 3.1 é um exemplo simplificado de um componente que referencia uma imagem retornada de uma busca. A linha dois do código define o componente que contém a URL da imagem, mas é só na linha três que o componente, que de fato exibe a imagem, é descrito. A imagem referenciada por este componente é representada por uma codificação base 64 e este código está na propriedade *src* do componente da imagem.

```

1 <div class="rg_di rg_bx rg_el ivg-i">
2   <a href="/imgres?imgurl=URL_DA_IMAGEM">
3     
4   </a>
5 </div>
```

Código 3.1 – Exemplo Simplificado de Componente *HTML*

Se habilitado no *browser* pelo usuário, os códigos *JavaScript* das páginas na *Web* serão executados. Estes códigos podem estar contidos na própria página a partir dos desenvolvedores da página ou podem ser inseridos de acordo com que as páginas são carregadas no navegador, isto acontece através da ação do próprio navegador ao suportar as extensões.

Sempre que uma página é completamente carregada, um evento é disparado e pode ser identificado por aplicações *JavaScript*. Quando isto acontece, a aplicação desenvolvida neste trabalho varre os elementos HTML em busca de componentes que são da classe *rg_ic rg_i*. Esta classe identifica as imagens que são retornadas da busca como mostra o Código 3.1. A extensão identifica que a página atual contém resultados de busca através do Algoritmo 7.

Algoritmo 7 Função que verifica se página em questão é página de resultados da busca por imagens.

```
1: função ISRESULTPAGE(document)
2:   imagens ← document.getElementsByClassName(“rg_ic rg_i”)
3:   imagesLength ← imagens.length
4:   se imagensLength > 0 então
5:     retorne true
6:   senão
7:     retorne false
8:   fim se
9: fim função
```

Uma vez que as imagens, que serão a entrada do algoritmo são identificadas na página, o algoritmo segue para a fase de geração dos descritores. Nesta fase, a geração do descritor acontece através de um código em *JavaScript* que interage com o componente HTML *canvas* que permite o acesso aos *pixels* das imagens.

A aplicação suporta o uso dos dois descritores apresentados. A razão para utilização de dois descritores diferentes é que estes são usados pelo algoritmo de agrupamento e dependendo da escolha do descritor, o algoritmo pode apresentar melhor ou pior qualidade em seus resultados.

3.2 Carregamento de Imagens e Geração de Descritores

A página de busca de imagens do *Google* pode retornar diversas imagens que vão sendo exibidas de acordo com que o usuário rola a página para baixo. Os componentes que contém as imagens indicam se a imagem está carregada ou não através do atributo *src* que significa *source* e indica a origem da imagem, que pode ser uma URL ou o próprio conteúdo codificado da imagem.

Uma vez que a extensão identifica que o documento carregado é identificado como a página de resultados de busca, a aplicação filtra as imagens que foram carregadas com base

no atributo *source* descrito anteriormente. Estas imagens são adicionadas a uma estrutura que referenciará as imagens carregadas.

Quando o descritor em questão é o autovetor, a aplicação varre as imagens identificando as imagens não quadradas, ou seja imagens cuja altura e largura são diferentes, para que estas sejam transformadas em imagens quadradas a fim de que seja possível calcular o autovetor dominante da imagem em questão. Isto é feito adicionando elementos à dimensão que possui menos elementos. Por exemplo, seja a imagem $A_{n \times m}$ de forma que $n > m$, neste caso a imagem possui mais linhas que colunas e por isso $n - m$ colunas serão adicionadas para que a imagem seja quadrada.

Uma vez que a etapa de carregamento das imagens é finalizada, a etapa de geração de descritores varre todas as imagens carregadas acessando os *pixels* através do componente *canvas*. O *canvas* é um componente HTML usado para desenhar gráficos e animações via *scripts*. O HTML *canvas* oferece uma série de opções para gerenciamento da cena.

O *canvas* possui uma referência ao objeto *context* que oferece métodos para o gerenciamento do *canvas*. É possível referenciar o *context* através do método *getContext()* e assim acessar métodos como o *drawImage()* que desenha um componente de imagem, vídeo ou próprio *canvas* passado como parâmetro.

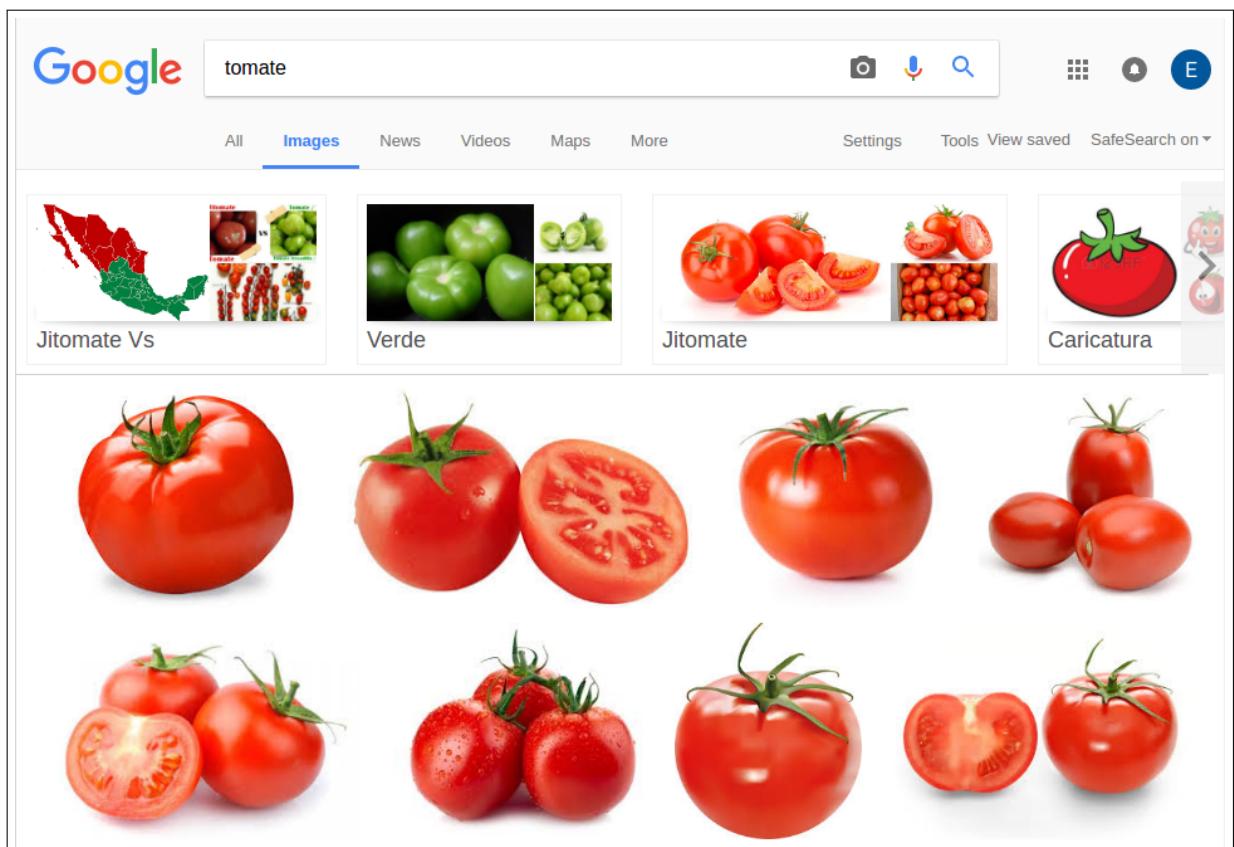
Este componente suporta a exibição de imagens suportadas pelo próprio navegador, desta forma é desnecessário qualquer tipo de tratamento ou conversão quanto aos tipos de imagens. Com isso, é possível acessar cada *pixel* de qualquer imagem que esteja sendo exibida na página e assim calcular o descritor em questão da respectiva imagem.

3.3 Agrupamento e Exibição dos Resultados

O Algoritmo 5 é implementado em *JavaScript* e o mesmo código é usado independentemente do tipo de descritor em questão. Isto porquê uma vez que os descritores são gerados, o algoritmo só precisa calcular as distâncias. A aplicação armazena o resultado do algoritmo em uma matriz, de forma que a quantidade de linhas representa o número de *clusters* e as colunas representam os elementos.

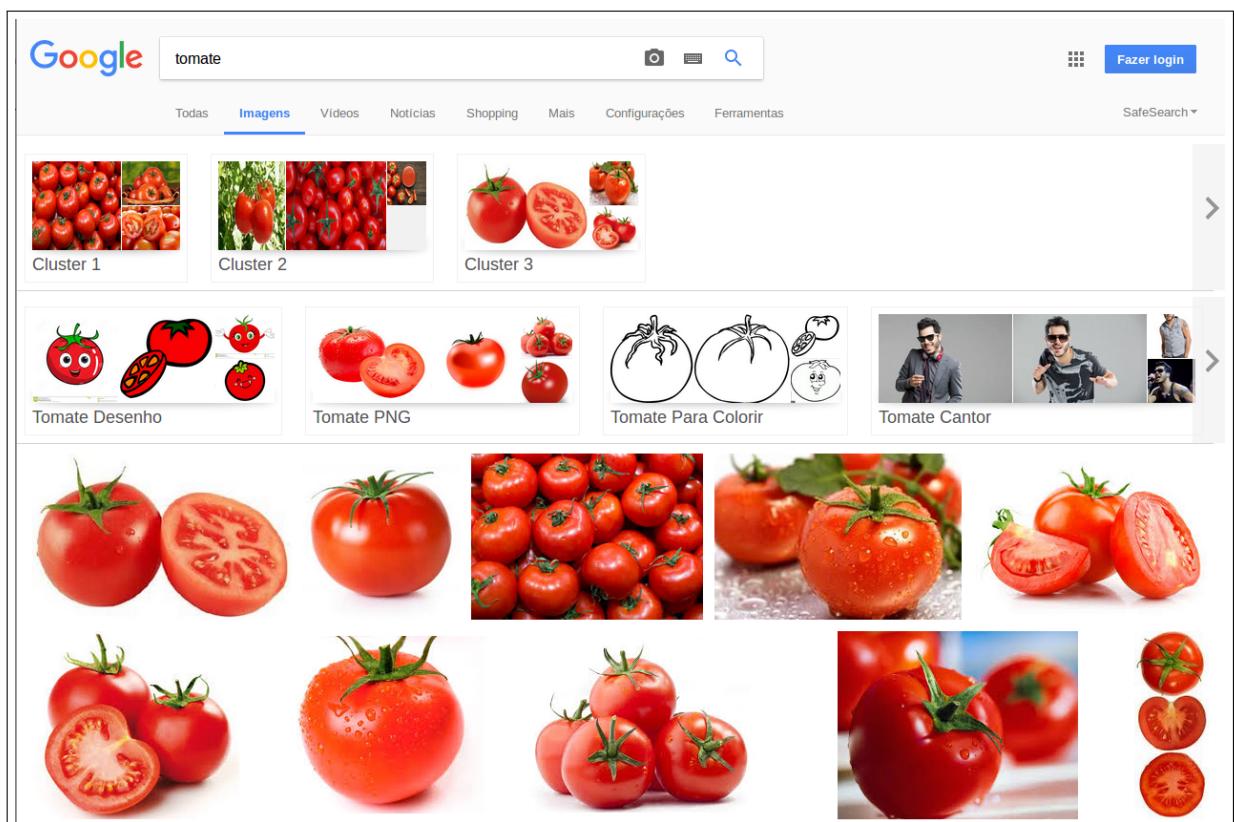
A apresentação dos resultados acontece na própria página de resultados. Algumas páginas de resultados do *Google* possuem um componente HTML que exibe grupos relacionados a busca feita. A Figura 11 mostra a região que classifica a busca por tomate em Jitomate Vs, Verde, Jitomate e Caricatura. Esta mesma região será duplicada na página para exibir os resultados do algoritmo. Uma vez que esta região é copiada na página, a aplicação substitui as imagens pelas imagens agrupadas pela aplicação como mostra a Figura 12.

Figura 11 – Resultado de busca.



Fonte: Google Imagens (2017a).

Figura 12 – Exibição do resultado do algoritmo na própria página.



Fonte: Google Imagens (2017a).

4 Resultados

Este capítulo apresenta os resultados obtidos com a aplicação desenvolvida e uma discussão acerca destes. A análise dos resultados será feita através da comparação dos dois descritores implementados. Antes de apresentar os resultados com imagens provenientes de uma busca, serão apresentados resultados do algoritmo em uma página de testes. As páginas de testes foram usadas como guias de ajustes e validação visual do algoritmo proposto.

A Tabela 1 apresenta as imagens presentes em uma página de teste. Apesar de todas estas imagens serem diferentes, as características que elas carregam são semelhantes entre algumas. As imagens da linha um são de tomates apresentados em diferentes ângulos em uma imagem de fundo branco. Semelhantemente, a linha dois apresenta imagens de balões com fundo azul não necessariamente na mesma tonalidade.

Tabela 1 – Imagens de página de teste.

		
1	2	3
		
4	5	6

Fonte: Google Imagens (2017a), Google Imagens (2017b).

A matriz de distâncias entre as imagens calculada através dos autovetores dominantes é apresentada na Tabela 2 e a calculada através dos histogramas é apresentada na Tabela 3. As células destacadas indicam os dois menores valores, ou seja as duas imagens mais próximas.

Tabela 2 – Matriz de distâncias - Autovetores Dominantes.

	1	2	3	4	5	6
1	0	0.7903890183496494	1.267739792773765	1.5707963267948966	1.5707960225553694	1.5707963267027802
2	0.7903890183496494	0	1.1557908824986878	1.5707963267948966	1.5707962171666512	1.5707963266958864
3	1.267739792773765	1.1557908824986878	0	1.5707963267948966	1.5632618141959285	1.5707963267041545
4	1.5707963267948966	1.5707963267948966	1.5707963267948966	0	1.5707962924625574	1.5707804758766788
5	1.5707960225553694	1.5707962171666512	1.5632618141959285	1.5707962924625574	0	1.5707679774026786
6	1.5707963267027802	1.5707963266958864	1.5707963267041545	1.5707804758766788	1.5707679774026786	0

Fonte: Autoria Própria

Tabela 3 – Matriz de distâncias - Histogramas.

	1	2	3	4	5	6
1	0	0.5756075471698113	0.5971680480729381	0.6794026881720429	0.6567888888888888	0.7045434782608694
2	0.5756075471698113	0	0.42156721584082	0.6947065074051536	0.7187795946890287	0.7458547990155866
3	0.5971680480729381	0.42156721584082	0	0.6477030282876355	0.588672665039677	0.6363063718865803
4	0.6794026881720429	0.6947065074051536	0.6477030282876355	0	0.4964178912783750	0.2739817671809256
5	0.6567888888888888	0.7187795946890287	0.588672665039677	0.4964178912783750	0	0.3671427536231884
6	0.7045434782608694	0.7458547990155866	0.6363063718865803	0.2739817671809256	0.3671427536231884	0

Fonte: Autoria Própria

Como esperado, o algoritmo de agrupamento por nuvens dinâmicas alocou todas as imagens da linha um (tomates) em um grupo e consequentemente as imagens da linha dois (balões) foram alocadas em um grupo distinto.

A seguir são apresentados os resultados da aplicação e da busca pelas palavras chaves, casa, lentilha e gato. As Tabelas 4, 10, 7 e 13 apresentam as imagens na ordem inicial disposta na página do *Google*. As outras tabelas apresentam os resultados obtidos através da extensão e dispõem as imagens de acordo com as alocações de forma que cada agrupamento é identificado por uma cor de fundo. Cada imagem é identificada por um número e os protótipos de cada grupo são identificados pelo número de identificação entre colchetes.

A Tabela 5 apresenta o conjunto de imagens da Tabela 4 organizado em três grupos. É possível observar que neste conjunto existem imagens similares como 1, 10, 13, 11, 21 que foram alocadas no mesmo grupo. De maneira semelhante, as imagens 0 e 8 foram alocadas no mesmo grupo. Entretanto, também é possível observar que as imagens 25 e 32 foram alocadas em grupos diferentes sendo que elas são visualmente parecidas, apesar de uma ser colorida e a outra não. Isto pode acontecer devido a diferenças como contraste por exemplo.

O resultado para o conjunto de imagens da Tabela 4 em três grupos usando os histogramas como descritores está disposto na Tabela 6. Com o uso de histogramas como descritores, parecem mais heterogêneos se comparados aos grupos da Tabela 5. Desta forma, para este conjunto de dados, o uso dos autovetores dominantes parece mais adequado.

As imagens da Tabela 7 apresentam o conjunto de imagens obtido na pesquisa pela palavra lentilha. Este conjunto foi organizado em quatro grupos. Neste conjunto é possível observar várias imagens com regiões das cores marrom, branca e bege. Os resultados usando os autovetores dominantes são apresentados na Tabela 8 e os resultados usando os histogramas estão na Tabela 6.

O uso dos histogramas pareceu mais adequado para o conjunto de dados da Tabela 7 uma vez que a Tabela 9 apresenta imagens melhores distribuídas de acordo com os aspectos visuais. No grupo um é possível ver imagens com predominância do branco e com cores concentradas no centro das imagens enquanto que o grupo dois apresenta imagens com cores mais distribuídas em meio ao branco assim como o grupo quatro. O grupo três concentra as imagens com regiões de cor marrom representadas pela imagem 26 que é o protótipo do grupo.

As imagens da Tabela 10 foram divididas em cinco grupos. Neste conjunto de dados é possível observar algumas imagens com características visualmente semelhantes, como as imagens 0, 2, 5, 6, 10, 11, 16, 30, 31, 32 que possuem fundo branco. Além dessas imagens é possível observar que apesar de algumas imagens não possuírem fundo branco, têm o branco predominando.

A Tabela 11 mostra os resultados da aplicação ao usar os autovetores dominantes como descritores. É possível observar que o primeiro grupo é homogêneo, mas poderia incluir outras

imagens com as mesmas características como 16, 6, 2, 5, 10 e 30. Os elementos do segundo grupo parecem similares ao possuírem regiões verdes e regiões escuras. O terceiro grupo apresenta o elemento 28 que visualmente se encaixaria na característica tida como dominante do grupo dois. Apesar disso, as outras imagens do grupo se assemelham assim como as imagens dos próximos grupos.

A Tabela 12 apresenta os resultados usando os histogramas como descritores e assim como na Tabela 11, algumas imagens com características predominantes brancas foram agrupadas em grupos diferentes. Entretanto isso não significa que os grupos estão heterogêneos como em alguns grupos dos resultados do outro descritor, mas que os grupos 1, 3 e 5 são semelhantes. Os grupos dois e quatro são homogêneos, o dois ao apresentar claramente imagens predominantemente com regiões escuras não encontradas em outras imagens do conjunto.

O conjunto de imagens da Tabela 13 apresenta o conjunto de imagens obtidas através da busca por gato que foi dividido em seis grupos. Para este conjunto, os resultados usando os autovetores dominantes e histogramas estão dispostos na Tabela 14 e Tabela 15 respectivamente.

Para o conjunto de dados da Tabela 13 os resultados com os autovetores dominantes se apresentaram mais satisfatórios se comparados aos resultados do outro descritor. O fato é que os resultados da Tabela 15 não se apresentam criteriosos nas divisões apresentadas enquanto que os resultados da Tabela 14 estão melhores distribuídos.

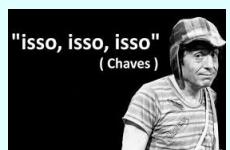
O grupo um da Tabela 14 apresenta imagens com aspectos mais neutros se comparadas as outras imagens do mesmo conjunto, mas é possível observar que o restante dos grupos apresenta resultados mais semelhantes. O grupo dois contém imagens com tons de cinza com baixo contraste, em contrapartida o grupo cinco dispõe de imagens com mais contraste. O grupo quatro destes resultados se destaca ao ter todas as imagens com gatos alaranjados. Além do grupo quatro, o grupo seis também se mostra homogêneo ao dispor de imagens de gatos ao centro com fundo predominantemente branco.

Tabela 4 – Imagens de página de pesquisa - Chaves.

	0		1		2
	3		4		5
	6		7		8
	9		10		11
	12		13		14
	15		16		17
	18		19		20
	21		22		23
	24		25		26
	27		28		29
	30		31		32

Fonte: Google Imagens (2017c).

Tabela 5 – Autovetor - Resultado do conjunto chaves em 3 grupos.

	0		8		20
	26		28		30
	31		1		3
	4		7		9
AVES 	10		11		12
	13		15		18
	21		22		23
	24		25		28
	29		2		5
	6		14		16
	17		19		32

Fonte: Google Imagens (2017c).

Tabela 6 – Histograma - Resultado do conjunto chaves em 3 grupos.

	0		29		30
	8		31		2
	4		5		3
	6		7		9
	1		16		12
	17		19		27
	18		10		11
	13		14		15
	20		21		22
	23		24		26
	25		28		32

Fonte: Google Imagens (2017c).

Tabela 7 – Imagens de página de pesquisa - Lentilha.

	0		1		2
	3		4		5
	6		7		8
	9		10		11
	12		13		14
	15		16		17
	18		19		20
	21		22		23
	24		25		26
	27		28		29
	30		31		32

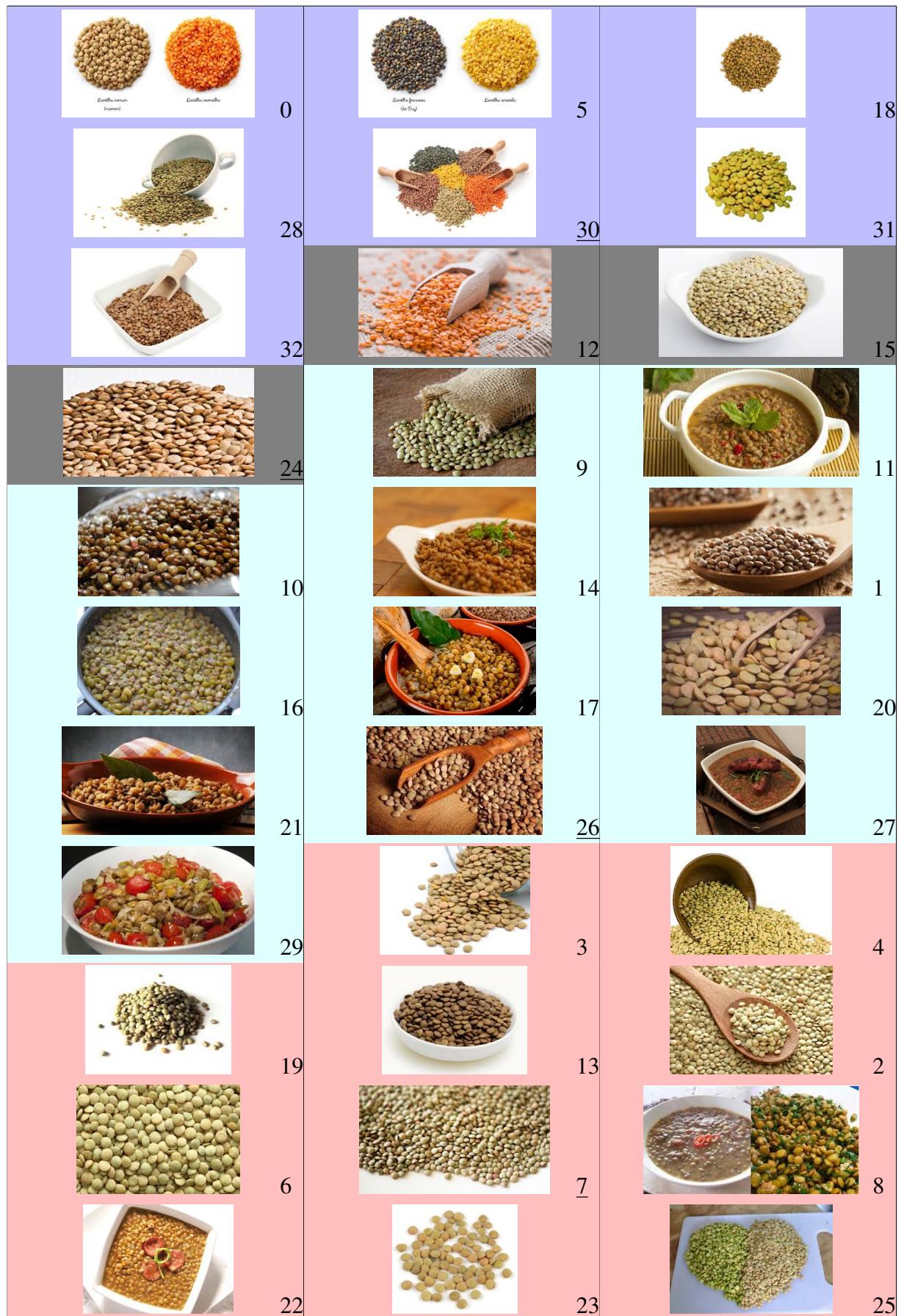
Fonte: Google Imagens (2017d).

Tabela 8 – Autovetor - Resultado do conjunto lentilha em 4 grupos.

	10		11		14
	28		30		31
	32		1		0
	15		17		21
	24		27		29
	2		4		25
	3		8		12
	13		18		6
	19		20		22
	23		5		16
	26		9		7

Fonte: Google Imagens (2017d).

Tabela 9 – Histograma - Resultado do conjunto lentilha em 4 grupos.



Fonte: Google Imagens (2017d).

Tabela 10 – Imagens de página de pesquisa - Casa.

	0		1		2
	3		4		5
	6		7		8
	9		10		11
	12		13		14
	15		16		17
	18		19		20
	21		22		23
	24		25		26
	27		28		29
	30		31		32

Fonte: Google Imagens (2017e).

Tabela 11 – Autovetor - Resultado do conjunto casa em 5 grupos.

0	11	31
32	1	14
25	27	13
28	29	3
16	17	18
20	22	23
26	6	2
5	10	30
4	8	7
12	15	19
21	24	9

Fonte: Google Imagens (2017e).

Tabela 12 – Histograma - Resultado do conjunto casa em 5 grupos.



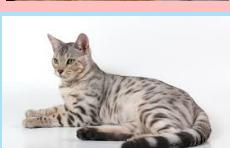
Fonte: Google Imagens (2017e).

Tabela 13 – Imagens de página de pesquisa - Gato.

	0		1		2
	3		4		5
	6		7		8
	9		10		11
	12		13		14
	15		16		17
	18		19		20
	21		22		23
	24		25		26
	27		28		29
	30		31		32

Fonte: Google Imagens (2017f).

Tabela 14 – Autovetor - Resultado do conjunto Gato em 6 grupos.

	6		7		17
	24		27		29
	9		20		22
	23		32		21
	16		26		3
	11		28		8
	0		18		12
	14		15		31
	5		4		19
	2		1		13
	18		25		30

Fonte: Google Imagens (2017f).

Tabela 15 – Histograma - Resultado do conjunto Gato em 6 grupos.

	6		7		17
	22		23		26
	1		19		27
	30		8		4
	5		13		21
	18		16		2
	9		25		10
	0		11		14
	20		3		24
	28		29		31
	15		12		32

Fonte: Google Imagens (2017f).

5 Conclusão e Trabalhos Futuros

Diante da necessidade de organização de grandes volumes de imagens, este trabalho propôs uma aplicação que auxilia o usuário em suas buscas por imagens ao usar o serviço de busca de imagens do *Google*. A aplicação desenvolvida possibilita que as imagens retornadas sejam dispostas em grupos que têm imagens semelhantes entre si. Desta forma, o usuário pode ter uma visão sistemática do resultado obtido, evitando assim, a necessidade de se visualizar todas as imagens quando se busca por uma específica. O *software* desenvolvido corresponde a pesquisas iniciais na área e pode ser aperfeiçoado se implementado e disponibilizado em servidores dedicados a este tipo de serviço.

Os resultados se mostraram satisfatórios dependendo do descritor adotado. O uso do autovetor dominante pareceu apropriado para os conjuntos de dados das Tabelas 4 e 13, como descrito no capítulo quatro. Já o uso do histograma pareceu adequado para as imagens das Tabelas 9 e 10. O fato é que a eleição de um descritor como melhor não é possível uma vez que cada conjunto de imagens é peculiar nas características mantidas pelo descritor, peculiar também assim como a expectativa do usuário.

A qualidade dos resultados obtidos pela aplicação também depende que a quantidade de grupos escolhidas por usuário seja adequada à configuração dos elementos. No entanto, a tarefa de escolher uma quantidade de grupos adequada pode ser subjetiva para alguns conjuntos de imagens. Então a elaboração de algum mecanismo que indique uma quantidade de grupos adequada ao conjunto de dados aprimoraria a qualidade da aplicação.

Como trabalhos futuros pretende-se a elaboração de meios para validação ou avaliação dos resultados obtidos. Com um método de avaliação dos resultados seria possível mensurar a qualidade dos descritores e ainda do algoritmo proposto. Além disso, trabalhos futuros estão relacionados com a utilização de outros descritores para imagens. Outra abordagem está relacionada também com a combinação de descritores que correspondem a um conjunto de descritores que otimize a separação das imagens. Propõe-se ainda a utilização de outros conjuntos de dados provenientes de outros acervos e serviços para que sejam usados como dados de entrada. Acervos com diferenças mais significativas entre as imagens podem oferecer uma melhor qualidade nos resultados.

Referências

- BERRY, M. J.; LINOFF, G. *Data mining techniques: for marketing, sales, and customer support.* [S.l.]: John Wiley & Sons, Inc., 1997.
- BOLDRINI, J. L. et al. *Álgebra Linear.* 3rd. ed. Brazil: HARBRA EDITORA, 1986. ISBN 8529402022.
- BOVIK, A. C. *Handbook of Image and Video Processing (2nd Edition).* Amsterdam: Elsevier, Inc., 2005. ISBN 0121197921, 978-0121197926.
- CARVALHO, F. de A. T. de; LECHEVALLIER, Y.; de Souza, R. M. C. R. A Dynamic Cluster Algorithm Based on L_r Distances for Quantitative Data. In: *Classification, Clustering, and Data Mining Applications.* Berlin, Heidelberg: Springer, 2004. p. 33–31.
- CARVALHO, L. de S. *Agrupamento e Regressão Linear de Dados Simbólicos Intervalares Baseados Em Novas Representações.* 2016.
- COULOURIS, G. et al. *Sistemas Distribuídos : Conceitos e Projeto.* 5th. ed. [S.l.]: Bookman, 2013. ISBN 978-0-13-214301-1.
- DEZA, M. M.; DEZA, E. *Encyclopedia of Distances.* [S.l.]: Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00233-5.
- EAKINS, J.; GRAHAM, M. Content-based image retrieval. Education-line, 1999.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2008. ISBN 013168728X.
- Google. *From the garage to the Googleplex.* 2017. [Online; acessado em 6 de Abril de 2017]. Disponível em: <<https://www.google.com/intl/en/about/our-story/>>.
- Google Imagens. 2017a. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=tomate>>.
- Google Imagens. 2017b. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=balao>>.
- Google Imagens. 2017c. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=chaves>>.
- Google Imagens. 2017d. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=lentilha>>.
- Google Imagens. 2017e. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=casa>>.
- Google Imagens. 2017f. [Online; acessado em 2 de Maio de 2017]. Disponível em: <<https://www.google.com/search?safe=active&tbo=isch&q=gato>>.

- HARTVEDT, C. Using context to understand user intentions in image retrieval. In: IEEE. *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on.* [S.l.], 2010. p. 130–133.
- JAIN, A. K. *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN 0-13-336165-9.
- JAIN, M.; SINGH, S. A survey on: content based image retrieval systems using clustering techniques for large data sets. *International Journal of Managing Information Technology*, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 4, p. 23, 2011.
- LI, Z.-N.; DREW, M. S.; LIU, J. *Fundamentals of Multimedia*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 3319052896, 9783319052892.
- MONTABONE, S. *Installing OpenCV 2.4.9 in Ubuntu 14.04 LTS*. 2007. [Online; acessado em 4 de Abril de 2017]. Disponível em: <<http://www.samontab.com/web/>>.
- Mozilla Developer Network. *Web technology for developers*. 2017. [Online; acessado em 6 de Abril de 2017]. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web>>.
- Net Market Share. *Desktop Search Engine Market Share*. 2017. [Online; acessado em 7 de Abril de 2017]. Disponível em: <<https://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0&qpsp=2016&qpnnp=2&qptimeframe=Y>>.
- SAAD, Y. *Numerical methods for large eigenvalue problems*. 2nd. ed. [S.l.]: Society for Industrial and Applied Mathematics, 2011. ISBN 978-1-611970-72-2.
- TANENBAUM, A. S.; STEEN, M. van. *Distributed Systems : Principles and Paradigms*. 2nd. ed. [S.l.]: Pearson Education Inc., 2007. ISBN 978-85-7605-142-8.
- TUYTELAARS, T.; MIKOLAJCZYK, K. et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, Now Publishers, Inc., v. 3, n. 3, p. 177–280, 2008.
- W3C. *Architecture of the World Wide Web, Volume One*. 2004. [Online; acessado em 5 de Abril de 2017]. Disponível em: <<https://www.w3.org/TR/webarch/>>.
- WIDROW, B. A study of rough amplitude quantization by means of nyquist sampling theory. *IRE Transactions on Circuit Theory*, IEEE Computer Society Press, v. 3, n. 4, p. 266–276, dez. 1956. ISSN 0096-2007. Disponível em: <<http://ieeexplore.ieee.org/document/1086334/>>.
- ZHANG, P. *Iterative Methods for Computing Eigenvalues and Exponentials of Large Matrices*. 2009.