

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA/ELETRÔNICA  
CURSO SUPERIOR DE ENGENHARIA DE COMPUTAÇÃO

LUCAS DEL CASTANHEL DIAS  
SOLANO ALVES DE AGUIAR

**USO DE ONTOLOGIAS COMO APOIO AO PROCESSO DE  
RECUPERAÇÃO DE IMAGEM POR CONTEÚDO BASEADO EM  
DESCRITORES**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2013

LUCAS DEL CASTANHEL DIAS  
SOLANO ALVES DE AGUIAR

**USO DE ONTOLOGIAS COMO APOIO AO PROCESSO DE  
RECUPERAÇÃO DE IMAGEM POR CONTEÚDO BASEADO EM  
DESCRITORES**

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Informática/Eletrônica como requisito parcial para obtenção do grau de Engenheiro no Curso Superior de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dra. Leyza Elmeri Baldo Dorini

Co-orientador: Prof. Dr. Cesar Augusto Tacla

**CURITIBA**

**2013**

Dedicamos este trabalho às nossas famílias, professores e amigos que nos acompanharam durante esta longa (e ao mesmo tempo curta) jornada de 5 anos de graduação.

## **AGRADECIMENTOS**

Com grande satisfação agradecemos nesta página todas as pessoas que contribuíram diretamente na elaboração deste trabalho.

Agradecemos à nossa orientadora, a Professora Dra. Leyza Baldo Dorini por seus conselhos, apoio e dedicação. À ela devemos muito. Somos muito gratos a seu trabalho e seus ensinamentos.

Agradecemos ao nosso co-orientador, o Professor Dr. César Augusto Tacla, o qual nos acompanhou durante quase todo o curso, sendo este trabalho, uma concretização do que com ele aprendemos.

Agradecemos à Vânia Del Castanhel, Luiza Del Castanhel, Célia Del Castanhel, Zalina Coan e Laura Del Castanhel Dias, mãe, tias, avó e irmã do Lucas, todas professoras, por todo o apoio, não só neste trabalho, mas durante toda a graduação.

Agradecemos à Talita Orlando de Campos, mulher do Solano, por todo o apoio e paciência durante toda a graduação e principalmente neste trabalho. Também pelo apoio da família do Solano: Solano pai, Lili mãe e Rodrigo e Murilo irmãos.

Agradecemos aos demais familiares do Lucas: os tios José, Ângelo, Vânio, Sônia e João Del Castanhel, que muito torceram durante estes cinco anos por cada aprovação de disciplinas.

Agradecemos a todos os professores do curso, por seus ensinamentos.

Agradecemos aos colegas de curso que nos ajudaram durante a graduação em todas as disciplinas que cursamos, em especial à Georgea Danielewicz, que nos auxiliou na leitura de algumas partes do texto.

There is something about Munch's "The Scream" or Constable's "Wivenhoe Park" that no words can convey. It has to be seen. The same holds for of a picture of the Kalahari Desert, a dividing cell, or the facial expression of an actor playing King Lear. It is beyond words. (...) Pictures have to be seen and searched as pictures: by objects, by style, by purpose. (SMEULDERS et al., 2000)

Há algo em "o Grito" de Munch ou em "Wivenhoe Park" de Constable que as palavras não convém. É algo que deve ser visto. O mesmo acontece para uma figura do deserto de Kalahari, a divisão de uma célula ou a expressão facial de um ator interpretando o rei Lear. Está além das palavras. (...). As figuras devem ser vistas e pesquisadas como figuras : por objetos, por estilo, por propósito. (SMEULDERS et al., 2000)

## RESUMO

DEL CASTANHEL, Lucas e AGUIAR, Solano Alves de. Uso de ontologias como apoio ao processo de recuperação de imagem por conteúdo baseado em descritores. 135 f. Trabalho de Conclusão de Curso – Curso Superior de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Fatores tais como a popularização dos computadores e da Internet, aliados ao fácil acesso e baixo custo de dispositivos digitais de captura de imagens, levaram a um crescimento exponencial das bases de dados contendo imagens. Associados à essa enorme fonte de informações visuais, surgiram problemas relacionados à busca desse conteúdo, ou seja, proprietários dos repositórios de imagens ou clientes de ferramentas agregadoras desse conteúdo passaram a demandar soluções de rápido resultado quanto à busca por determinados padrões visuais. Para atender à esta crescente demanda surgiu a recuperação de imagens baseada em conteúdo (do inglês *Content Based Image Retrieval* ou CBIR). Esta consiste na extração de características de cor, textura e forma das imagens e na aplicação de relações matemáticas que permitam compará-las entre si no intuito de realizar buscas. Contudo, esta metodologia esbarra em problemas semânticos : A representação das características das imagens e sua comparação não é intuitiva ao usuário final, o que dificulta a relação entre os resultados da busca e suas expectativas. Visando reduzir este problema, propõe-se utilizar ontologias para apoiar o processo de recuperação baseado em descritores para tornar o processo de busca e seus resultados mais intuitivos. Apresenta-se também um *software* de apoio à realização de experimentos e que pode ser expandido para outras abordagens. Utilizou-se uma base de imagens composta de espécies de flores para analisar o processo de recuperação baseado em descritores e propor o uso de ontologias para melhorar o processo de busca.

**Palavras-chave:** CBIR, descritor, ontologia

## ABSTRACT

DEL CASTANHEL, Lucas e AGUIAR, Solano Alves de. Ontologies as a support for content-based image retrieval based on descriptors. 135 f. Trabalho de Conclusão de Curso – Curso Superior de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Facts such as the spread of personal computers and Internet, allied with easy-access and low-cost digital image capture devices, have led to an exponential growth on image databases. Along this huge source of visual information, problems related to the queries of this content arose. The owners of the image repositories or clients of the content's tools became to demand faster solutions when querying for some certain visual patterns. With the purpose of fulfilling this increasing demand, the CBIR (Content Based Image Retrieval) was designed. The latter consists on the extraction of color, shape and texture features from images and the application of mathematical relations which enables the comparison between images, aiming to respond to queries. However, this methodology faces semantic problems: The representation of the image features and its comparison is not intuitive to the user, which disturbs the relationship between the query results and their proper outcome. Aiming to handle this problem, we propose the use of ontologies to support the descriptor-based retrieval process, in order to make the query process more intuitive. In addition, we introduce an experiment-driven software, which might be expanded to another approaches. The chosen image dataset is comprised of flower species and has been used to analyze the descriptor-based query process and to propose how an ontology might improve it.

**Keywords:** CBIR, descriptor, ontology

## LISTA DE FIGURAS

FIGURA 1 – O DESCRITOR DE IMAGENS .....	22
FIGURA 2 – EXEMPLO DE HISTOGRAMA GLOBAL DE COR PARA TRÊS IMA- GENS .....	26
FIGURA 3 – DESCRITOR DE COR DOMINANTE .....	27
FIGURA 4 – DESCRITOR DE FEIXO CONVEXO .....	28
FIGURA 5 – DFC - EXTRAÇÃO DO DESVIO PADRÃO .....	29
FIGURA 6 – DFC - DESVIO PADRÃO ENTRE PONTOS DO CONTORNO .....	29
FIGURA 7 – SIFT .....	30
FIGURA 8 – EXEMPLO DE CLASSIFICAÇÃO POR KNN .....	31
FIGURA 9 – EXEMPLO DE BUSCA DE IMAGENS ILUSTRANDO A TABELA DE CONTINGÊNCIA .....	33
FIGURA 10 – EXEMPLO DE CURVA PRECISÃO E REVOCAÇÃO .....	34
FIGURA 11 – CBIR : ARQUITETURA TÍPICA .....	36
FIGURA 12 – INTERFACE DE BUSCAS POR IMAGEM IMAGEM DO GOOGLE.	37
FIGURA 13 – EXEMPLO DE <i>SEMANTIC GAP</i> .....	39
FIGURA 14 – <i>SENSORIAL GAP</i> - UM DOS PROBLEMAS DAS TÉCNICAS DE CBIR	39
FIGURA 15 – ATIVIDADES DO METHONTOLOGY .....	43
FIGURA 16 – COMPONENTES DO METHONTOLOGY .....	44
FIGURA 17 – EXEMPLO DE ONTOLOGIA DE OBJETO .....	46
FIGURA 18 – PROCESSO DE MAPEAMENTO ENTRE ATRIBUTOS DE BAIXO NÍ- VEL PARA INDIVÍDUOS DA ONTOLOGIA DE OBJETO .....	47
FIGURA 19 – SEPARAÇÃO DAS CLASSES DE FLORES EM PASTAS .....	50
FIGURA 20 – ARQUIVO DE MARCAÇÃO XML DO EXPERIMENTO .....	51
FIGURA 21 – DIAGRAMA DE EXECUÇÃO DE EXPERIMENTO .....	52
FIGURA 22 – ARQUIVO DE MARCAÇÃO XML DO EXPERIMENTO .....	53
FIGURA 23 – ARQUIVO DE MARCAÇÃO XML DO EXPERIMENTO .....	54
FIGURA 24 – INTERFACE PARA VALIDAÇÃO DE DESCRITORES .....	55
FIGURA 25 – SAÍDA DO CÁLCULO DE DESCRITORES .....	55
FIGURA 26 – EXEMPLO DE EXECUÇÃO DE EXPERIMENTO .....	56
FIGURA 27 – TELA INICIAL DA FERRAMENTA .....	57
FIGURA 28 – EXEMPLO DE EXECUÇÃO DE EXPERIMENTO DE ONTOLOGIAS	58
FIGURA 29 – SOFTWARE NA PERSPECTIVA DO PADRÃO ARQUITETURAL <i>MO- DEL VIEW CONTROLLER</i> .....	58
FIGURA 30 – BASES CONTROLADAS .....	61
FIGURA 31 – BC 1 .....	62
FIGURA 32 – BC 2 .....	63
FIGURA 33 – BC 3 .....	64
FIGURA 34 – BC 4 .....	65
FIGURA 35 – BASE CONTROLADA N .....	66
FIGURA 36 – AMOSTRA CLASSIFICADA ERRONEAMENTE NO EXPERIMENTO	



1	68
FIGURA 37 – METODOLOGIA CBIR	94
FIGURA 38 – CURVA PRECISÃO VERSUS REVOCAÇÃO	95
FIGURA 39 – METODOLOGIA DE SISTEMA DE RECUPERAÇÃO DE IMAGENS	
COM ONTOLOGIA ADOTADA	96
FIGURA 40 – TAXONOMIA DE CONCEITOS DA ONTOLOGIA	98
FIGURA 41 – DIAGRAMA DE RELAÇÕES BINÁRIAS AD HOC	100
FIGURA 42 –	103
FIGURA 43 – METODOLOGIA DE BUSCAS ADOTADAS PARA A ONTOLOGIA	103
FIGURA 44 – PADRÃO TEMPLATE	111
FIGURA 45 – APRESENTAÇÃO DO PADRÃO DAO PARA GERAÇÃO, LEITURA E	
ESCRITA DE EXPERIMENTOS	112
FIGURA 46 – DIAGRAMA DE CASOS DE USO DO SISTEMA	114
FIGURA 47 – CLASSES DE DESCRITORES	120
FIGURA 48 – COMPOSIÇÃO DE UM EXPERIMENTO	121
FIGURA 49 – CLASSES DO EXPERIMENTO	122
FIGURA 50 – CLASSES DO EXPERIMENTO DA ONTOLOGIA	123
FIGURA 51 – SAÍDA DO EXPERIMENTO CBIR	123
FIGURA 52 – ACERTOS E ERROS POR CLASSE	123
FIGURA 53 – ARQUIVO DE PRECISÃO VS RECALL	124
FIGURA 54 – CRONOGRAMA - TAREFAS	126
FIGURA 55 – CRONOGRAMA - GANTT	127

## LISTA DE TABELAS

TABELA 1	–	EXEMPLO DE TABELA DE CONTINGÊNCIA	33
TABELA 2	–	EXEMPLO DE MATRIZ CONFUSÃO	35
TABELA 3	–	TOTAL DE IMAGENS POR CLASSE	61
TABELA 4	–	PRECISÃO CONSIDERANDO O DESCRITOR GCH NA BC 1	67
TABELA 5	–	MATRIZ DE CONFUSÃO PARA O DESCRITOR GCH NA BC 1	67
TABELA 6	–	PRECISÃO CONSIDERANDO O DESCRITOR DCD NA BC 1	68
TABELA 7	–	MATRIZ DE CONFUSÃO PARA O DESCRITOR DCD NA BC 1	68
TABELA 8	–	PRECISÃO CONSIDERANDO O DESCRITOR SIFT NA BC 1	69
TABELA 9	–	MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO SIFT NA BC 1	69
TABELA 10	–	PRECISÃO CONSIDERANDO O DESCRITOR DFC NA BC 1	70
TABELA 11	–	MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO DFC NA BC 1	70
TABELA 12	–	PRECISÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 1	70
TABELA 13	–	MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 1	70
TABELA 14	–	PRECISÃO CONSIDERANDO OS DESCRITORES GCH E SIFT NA BC 1	71
TABELA 15	–	MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES GCH E SIFT NA BC 1	71
TABELA 16	–	PRECISÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 1	72
TABELA 17	–	MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 1	72
TABELA 18	–	PRECISÃO CONSIDERANDO OS DESCRITORES DCD E SIFT NA BC 1	72
TABELA 19	–	MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES DCD E SIFT NA BC 1	73
TABELA 20	–	PRECISÃO CONSIDERANDO O DESCRITOR GCH NA BC 2	73
TABELA 21	–	MATRIZ DE CONFUSÃO PARA O DESCRITOR GCH NA BC 2	73
TABELA 22	–	PRECISÃO CONSIDERANDO O DESCRITOR DCD NA BC 2	74
TABELA 23	–	MATRIZ DE CONFUSÃO PARA O DESCRITOR DCD NA BC 2	74
TABELA 24	–	PRECISÃO CONSIDERANDO O DESCRITOR SIFT NA BC 2	75
TABELA 25	–	MATRIZ DE CONFUSÃO PARA O DESCRITOR SIFT NA BC 2	75
TABELA 26	–	PRECISÃO CONSIDERANDO O DESCRITOR DFC NA BC 2	75
TABELA 27	–	MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO DFC NA BC 2	75

	NA BC 2 .....	76
TABELA 28	– PRECISÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 2 .....	76
TABELA 29	– MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 2 .....	77
TABELA 30	– PRECISÃO CONSIDERANDO OS DESCRITORES GCH E SIFT NA BC 2 .....	77
TABELA 31	– MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES GCH E SIFT NA BC 2 .....	77
TABELA 32	– PRECISÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 2 .....	78
TABELA 33	– MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 2 .....	78
TABELA 34	– PRECISÃO CONSIDERANDO OS DESCRITORES DCD E SIFT NA BC 2 .....	78
TABELA 35	– MATRIZ DE CONFUSÃO CONSIDERANDO OS DESCRITORES DCD E SIFT NA BC 2 .....	78
TABELA 36	– PRECISÃO CONSIDERANDO O DESCRITOR GCH NA BC 3 ...	79
TABELA 37	– MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO GCH NA BC 3 .....	79
TABELA 38	– PRECISÃO CONSIDERANDO O DESCRITOR DCD NA BC 3 ...	80
TABELA 39	– MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO DCD NA BC 3 .....	80
TABELA 40	– PRECISÃO CONSIDERANDO O DESCRITOR SIFT NA BC 3 ....	81
TABELA 41	– MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO SIFT NA BC 3 .....	81
TABELA 42	– PRECISÃO CONSIDERANDO O DESCRITOR DFC NA BC 3 ....	82
TABELA 43	– MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO DFC NA BC 3 .....	82
TABELA 44	– PRECISÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 3 .....	83
TABELA 45	– MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES GCH E DFC NA BC 3 .....	83
TABELA 46	– PRECISÃO CONSIDERANDO OS DESCRITORES GCH E SIFT NA BC 3 .....	83
TABELA 47	– MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES GCH E SIFT NA BC 3 .....	83
TABELA 48	– PRECISÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 3 .....	84
TABELA 49	– MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES DCD E DFC NA BC 3 .....	84
TABELA 50	– PRECISÃO CONSIDERANDO OS DESCRITORES DCD E SIFT NA	

BC 3 .....	85
TABELA 51 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES DCD E SIFT NA BC 3 .....	85
TABELA 52 – PRECISÃO CONSIDERANDO O DESCRITOR GCH NA BC 4 ...	85
TABELA 53 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA O DESCRITOR GCH NA BC 4 .....	86
TABELA 54 – PRECISÃO CONSIDERANDO O DESCRITOR DCD NA BC 4 ...	86
TABELA 55 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA O DESCRITOR DCD NA BC 4 .....	86
TABELA 56 – PRECISÃO CONSIDERANDO O DESCRITOR SIFT NA BC 4 ....	87
TABELA 57 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA O DESCRITOR SIFT NA BC 4 .....	87
TABELA 58 – PRECISÃO CONSIDERANDO O DESCRITOR DFC NA BC 4 ....	88
TABELA 59 – MATRIZ DE CONFUSÃO DO EXPERIMENTO UTILIZANDO DFC NA BC 4 .....	88
TABELA 60 – PRECISÃO CONSIDERANDO OS DESCRITORES GCH E DFC NA BC 4 .....	89
TABELA 61 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES GCH E DFC NA BC 4 .....	89
TABELA 62 – PRECISÃO CONSIDERANDO OS DESCRITORES GCH E SIFT NA BC 4 .....	90
TABELA 63 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES GCH E SIFT NA BC 4 .....	90
TABELA 64 – PRECISÃO CONSIDERANDO OS DESCRITORES DCD E DFC NA BC 4 .....	91
TABELA 65 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES DCD E DFC NA BC 4 .....	91
TABELA 66 – PRECISÃO CONSIDERANDO OS DESCRITORES DCD E SIFT NA BC 4 .....	92
TABELA 67 – MATRIZ DE CONFUSÃO DO EXPERIMENTO PARA OS DESCRITORES DCD E SIFT NA BC 4 .....	92
TABELA 68 – INFORMAÇÕES SOBRE CADA CLASSE DE OBJETOS .....	97
TABELA 69 – GLOSSÁRIO DE CONCEITOS E INSTÂNCIAS DA ONTOLOGIA .....	99
TABELA 70 – GLOSSÁRIO DE RELAÇÕES DA ONTOLOGIA .....	100
TABELA 71 – DICIONÁRIO DE CONCEITOS .....	101
TABELA 72 – DEFINIÇÃO DAS RELAÇÕES <i>AD HOC</i> .....	101
TABELA 73 – DEFINIÇÃO DAS INSTÂNCIAS DO CONCEITO “PLANTA”, PARTE 1 .....	102
TABELA 74 – DEFINIÇÃO DAS INSTÂNCIAS DO CONCEITO “PLANTA”, PARTE 2 .....	104
TABELA 75 – DEFINIÇÃO DAS INSTÂNCIAS DOS CONCEITOS PAÍS E CONTINENTE .....	104

TABELA 76 – INFORMAÇÕES SOBRE CADA CLASSE DE OBJETOS	.....	106
TABELA 77 – VANTAGENS DE DESVANTAGENS DAS ABORDAGENS	.....	108

## LISTA DE SIGLAS

CBIR	Content Based Image Retrieval
W3C	World Web Consortium
GCH	<i>Global Color Histogram</i>
DCD	<i>Dominant Color Descriptor</i>
KNN	<i>K-Nearest Neighbors</i>
W3C	World Web Consortium
OWL	Web Ontology Language
<i>HOG</i>	Histogram of Gradients
UML	Unified Modeling Language
GoF	<i>Gang of Four</i>
DAO	Data access Object

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>17</b>
1.1 VISÃO GERAL DO TRABALHO	19
1.2 TRABALHOS RELACIONADOS	20
<b>2 RECUPERAÇÃO DE IMAGENS BASEADA EM CONTEÚDO</b>	<b>21</b>
2.1 DESCRITORES	21
2.1.1 Extração de características	22
2.1.2 Funções de distância	23
2.1.2.1 Distância L1	24
2.1.2.2 Distância L2	24
2.1.2.3 Distância $L_\infty$	24
2.2 DESCRITORES DE COR	24
2.2.1 Global Color Histogram - GCH	25
2.2.2 Dominant Color Descriptor	26
2.3 DESCRITORES DE FORMA	27
2.3.1 Descritor de Feixo Convexo	28
2.3.2 Scale-Invariant Feature Transform (SIFT)	29
2.4 CLASSIFICAÇÃO	30
2.4.1 K-Nearest Neighbors (KNN)	31
2.4.2 Medidores de desempenho da classificação	32
2.4.2.1 Precisão e revocação	32
2.4.2.2 Matriz de confusão	35
2.5 COMPONENTES BÁSICOS DE UM SISTEMA DE RECUPERAÇÃO DE IMAGENS POR CONTEÚDO	35
2.5.1 Interface	36
2.5.2 Módulo de processamento	37
2.5.3 Base de imagens	37
2.6 PROBLEMAS DE UM SISTEMA CBIR	38
2.6.1 <i>Semantic e sensorial gap</i>	38
2.7 CONSIDERAÇÕES SOBRE O CAPÍTULO	40
<b>3 ONTOLOGIAS</b>	<b>41</b>
3.1 ONTOLOGIAS	41
3.2 CONSTRUÇÃO DE ONTOLOGIAS	42
3.2.1 Conceitualização no <i>Methontology</i>	43
3.3 TRABALHOS RELACIONADOS ENVOLVENDO ONTOLOGIAS E CBIR	45
3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO	46
<b>4 FERRAMENTA DE SUPORTE AOS TESTES EXPERIMENTAIS</b>	<b>48</b>
4.1 MÓDULO DE PROCESSAMENTO	48
4.1.1 Criação de um experimento	49
4.1.2 Execução do experimento	50
4.1.3 Personalização das saídas do experimento	52
4.2 INTERFACE	52

4.2.1 Interface textual: experimentos .....	53
4.2.2 CBIR .....	54
4.2.3 Ontologia .....	55
4.3 BASE DE IMAGENS .....	55
4.4 VISÃO GERAL DOS COMPONENTES DO SOFTWARE .....	57
4.5 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	59
<b>5 DESENVOLVIMENTO .....</b>	<b>60</b>
5.1 BASE DE IMAGENS .....	60
5.1.1 Base Controlada 1 (BC 1) .....	60
5.1.2 Base Controlada 2 (BC 2) .....	61
5.1.3 Base Controlada 3 (BC 3) .....	62
5.1.4 Base Controlada 4 (BC 4) .....	62
5.2 VALIDAÇÃO DOS DESCRITORES .....	66
5.2.1 Avaliação dos descritores para classificação da BC 1 .....	67
Experimento 1 - GCH .....	67
Experimento 2 - DCD .....	68
Experimento 3 - SIFT .....	69
Experimento 4 - DFC .....	69
Experimento 5 - GCH e DFC .....	70
Experimento 6 - GCH e SIFT .....	70
Experimento 7 - DCD e DFC .....	72
Experimento 8 - DCD e SIFT .....	72
5.2.2 Avaliação dos descritores para classificação da BC 2 .....	72
Experimento 1 - GCH .....	73
Experimento 2 - DCD .....	74
Experimento 3 - SIFT .....	74
Experimento 4 - DFC .....	75
Experimento 5 - GCH e DFC .....	76
Experimento 6 - GCH e SIFT .....	76
Experimento 7 - DCD e DFC .....	77
Experimento 8 - DCD e SIFT .....	78
5.2.3 Avaliação dos descritores para classificação da BC 3 .....	79
Experimento 1 - GCH .....	79
Experimento 2 - DCD .....	79
Experimento 3 - SIFT .....	81
Experimento 4 - DFC .....	81
Experimento 5 - GCH e DFC .....	82
Experimento 6 - GCH e SIFT .....	82
Experimento 7 - DCD e DFC .....	83
Experimento 8 - DCD e SIFT .....	84
5.2.4 Avaliação dos descritores para classificação da BC 4 .....	84
Experimento 1 - GCH .....	85
Experimento 2 - DCD .....	86
Experimento 3 - SIFT .....	86
Experimento 4 - DFC .....	87
Experimento 5 - GCH e DFC .....	89
Experimento 6 - GCH e SIFT .....	90
Experimento 7 - DCD e DFC .....	91



Experimento 8 - DCD e SIFT .....	92
5.2.5 Considerações gerais.....	92
5.3 CBIR.....	93
5.3.1 Discussão dos resultados .....	94
5.4 METODOLOGIA PARA USO DE ONTOLOGIAS COMO APOIO AO UM SISTEMA DE RECUPERAÇÃO DE IMAGENS .....	96
5.4.1 Especificação da ontologia .....	97
5.4.2 Conceitualização .....	98
5.4.2.1 Glossário de termos .....	98
5.4.2.2 Taxonomias de conceito.....	98
5.4.2.3 Diagrama de relações binárias <i>ad hoc</i> .....	99
5.4.2.4 Dicionário de conceitos .....	100
5.4.2.5 Definição detalhada das relações <i>ad hoc</i> .....	100
5.4.2.6 Definição das instâncias.....	100
5.4.3 Implementação e Buscas na Ontologia.....	101
5.5 PROPOSTA DE USO DA ONTOLOGIA COMO APOIO AO PROCESSO DE RECUPERAÇÃO.....	105
5.6 CONSIDERAÇÕES SOBRE O ESTUDO TEÓRICO .....	107
<b>6 DOCUMENTAÇÃO DE SOFTWARE.....</b>	<b>109</b>
6.1 LITERATURA UTILIZADA NO PROJETO DE <i>SOFTWARE</i> .....	109
6.1.1 Requisitos .....	109
6.1.2 UML.....	109
6.1.3 Padrões de projeto .....	110
6.1.3.1 Padrão <i>Template</i> .....	110
6.1.3.2 Padrão DAO .....	110
6.2 PROJETO DE <i>SOFTWARE</i> .....	111
6.2.1 Levantamento de requisitos .....	111
6.2.1.1 Requisitos funcionais .....	112
6.2.2 Requisitos não funcionais .....	113
6.2.3 Casos de uso.....	113
6.2.3.1 Caso de uso: Criar Modelo de Experimento .....	114
6.2.3.2 Caso de uso: Executar Experimento CBIR .....	115
6.2.3.3 Caso de uso: Executar Experimento Ontologias .....	116
6.2.3.4 Caso de uso: Segmentar Base.....	117
6.2.3.5 Caso de uso: Calcular descritores .....	118
6.2.4 Diagramas de classes .....	118
6.2.5 Exemplo de saída de execução CBIR .....	121
6.3 CONSIDERAÇÕES SOBRE O CAPÍTULO .....	121
<b>7 GESTÃO DO PROJETO.....</b>	<b>125</b>
7.1 GESTÃO DE RECURSOS .....	125
7.1.1 Hardware.....	125
7.1.2 Software.....	125
7.1.3 Cronograma final.....	126
7.1.4 Considerações.....	128
<b>8 CONCLUSÃO.....</b>	<b>129</b>
8.1 CONTRIBUIÇÕES.....	130
8.2 TRABALHOS FUTUROS.....	130

<b>REFERÊNCIAS .....</b>	<b>132</b>
--------------------------	------------

## 1 INTRODUÇÃO

A popularização da internet, aliada ao fácil acesso a dispositivos de mídias digitais como câmeras, celulares e filmadoras, motivou o surgimento de extensas bases de dados de imagens que, entre diversas utilidades, auxiliam ao governo, instituições de ensino, indústrias e empresas em suas áreas de atuação (VELTKAMP; TANASE, 2000). Como procurar por imagens em tais bases pode ser custoso, técnicas robustas e eficientes de recuperação se fazem necessárias em meio ao crescimento explosivo da quantidade dessa mídia digital (YANG, 2004).

Tradicionalmente, as imagens armazenadas em bases de dados são indexadas por informação textual, atribuída através do processo de anotação, onde um usuário capacitado avalia e classifica manualmente cada imagem. No entanto, além de tal método ser altamente subjetivo (dependendo do usuário em questão), implica em um custo elevado. Além disso, existem também questões relacionadas à ambiguidade como, por exemplo, no caso de uma busca por imagens relacionadas à palavra “rosa” (isso refere-se à cor, à planta ou uma pessoa com esse nome?). Smeulders *et al.* (SMEULDERS et al., 2000) iniciam seu artigo com o seguinte argumento: “Há algo no quadro O grito de Munch que as palavras não conseguem representar”.

Desta forma, o ideal é a utilização de métodos automáticos para extração de características que descrevam cada imagem, diferenciando-a das demais. A área de pesquisa “recuperação de imagens por conteúdo” (em inglês, *Content Based Image Retrieval* - CBIR) envolve a definição de diferentes formas de indexação automática para recuperação de imagens. Cabe salientar que imagens são dados não-estruturados, consistindo de vetores com intensidades de pixel sem nenhum significado inerente. Uma das características de sistemas de CBIR é a de buscar extrair informações a partir das imagens, de forma a caracterizar seu conteúdo. Para isso, utilizam-se principalmente características primitivas como cor, textura e forma ou uma combinação destas (TORRES; FALCAO, 2005). CBIR tem seu termo originado a partir do trabalho de (KATO, 1992) para a automação de recuperação de imagens em base de dados baseada na análise de cor e forma presentes nas imagens (SINGHA; K.HEMACHANDRAN, 2012). Os sistemas de CBIR tem sido utilizados nas mais diversas áreas, dentre as quais medicina, entretenimento, bibliotecas digitais, sistemas de informação (ambiental, geográfica, entre outros) e identificação

de impressões digitais (TORRES; FALCAO, 2005).

Em resumo, tais sistemas tem por objetivo possibilitar a consulta a uma base de dados de imagens tendo como solicitação um padrão (outra imagem, um esboço, entre outros) fornecido pelo usuário. A consulta compara a similaridade entre as características da imagem de entrada e das imagens da base. Tais características são baseadas em conteúdo - cor, por exemplo - e estão organizadas em um vetor de características (denominados descritores), correspondente a uma “assinatura” da imagem. A extração das características é realizada de forma automática, e sua escolha depende da aplicação e da classe de imagens em questão (o atributo cor não é importante se estamos trabalhando com imagens em tons de cinza, por exemplo). Além disso, a utilização de medidas adequadas para avaliar a similaridade para cada descritor é fundamental na obtenção de bons resultados (YANG, 2004).

Embora diversos métodos robustos tenham sido propostos na literatura, apenas as características mencionadas tipicamente não são suficientes para representar a semântica da imagem. Neste contexto, a representação de imagens por meio de descritores possui várias limitações, dentre as quais a impossibilidade de representação de determinadas características (área, por exemplo, depende das condições de aquisição), a possibilidade de imagens não pertencentes à mesma classe possuírem descritores semelhantes (como a cor, por exemplo) e, analogamente, a possibilidade de imagens pertencentes à mesma classe possuírem descritores distintos. Note que problemas ocorrem quando há relação semântica envolvida (SHARMA; RAWAT, 2012). A não coincidência entre as representações dos sistemas de recuperação de imagens por conteúdo e da interpretação do usuário do sistema chama-se *semantic gap* (ou gap semântico) (WILSON; MARTINEZ, 1997).

Tais problemas originaram uma terceira classe de métodos de recuperação de imagens, baseada em ontologias (HYVÖNEN; STYRMAN; SAARELA, 2002). Neste caso, é utilizado um novo modo de anotação e recuperação com base em conceitos de alto nível (HYVÖNEN; STYRMAN; SAARELA, 2002). O consórcio W3C define a ontologia como “a definição dos termos utilizados na representação de uma área de conhecimento”.

Os sistemas de recuperação de imagens disponíveis na *web* dividem-se em duas principais categorias: aqueles onde o parâmetro de busca é representado por uma imagem e aqueles em que a busca é realizada através de elementos textuais extraídos de anotações manuais. Como discutido, ambos possuem limitações: enquanto as buscas metatextuais levam em conta apenas a anotação das imagens, a recuperação de imagens por conteúdo dispensa qualquer anotação.

Contudo, carecem estudos e ferramentas de *software* capazes de comparar as características complementares das abordagens por CBIR e ontologias considerando parâmetros tais como

eficiência e eficácia dos resultados de uma busca. Este é o ponto de partida deste trabalho, que tem como objetivo principal realizar um estudo de como estas duas abordagens podem ser utilizadas em conjunto, bem como a construção de uma ferramenta de software capaz de executar e analisar o desempenho das técnicas previstas no estudo teórico. Para validar a ferramenta e realizar o estudo de cada abordagem, foi utilizada parte da base de imagens apresentada em (NILSBACK; ZISSERMAN, 2012), a qual é constituída de 102 espécies de flores. Pretende-se, assim, explorar as vantagens de ambas as abordagens: recuperar imagens com características desejadas em grandes volumes de dados de forma automática.

## 1.1 VISÃO GERAL DO TRABALHO

O presente trabalho inicia no Capítulo 2 com a fundamentação teórica dos modelos de recuperação de imagens por conteúdo, explicando os conceitos de descritores de imagem, que são os elementos que caracterizarão digitalmente a imagem, funções de distância, classificação, precisão e revocação e componentes típicos de um sistema de CBIR. Os problemas da CBIR conhecidos como *Sensorial gap* e *Semantic gap* são brevemente explicados.

No Capítulo 3, são apresentados os conceitos relativos à ontologia, suas definições mais comuns da literatura. Discute-se também a respeito dos métodos de construção de uma ontologia, com ênfase no *Methontology* (CORCHO et al., 2005), adotado neste trabalho. Apresenta-se no final do capítulo a metodologia para um sistema de recuperação de imagens utilizando uma ontologia para reduzir o *semantic* e *sensorial gap*.

No Capítulo 4, apresenta-se a ferramenta desenvolvida para auxiliar a execução dos experimentos realizados neste trabalho. A discussão é apresentada nos moldes dos componentes típicos de um sistema de recuperação de imagens por conteúdo. O desenvolvimento do trabalho é apresentado no capítulo 5, dividido em três etapas: validação de descritores, experimento de recuperação baseado em descritores e avaliação da complementação com ontologias.

A documentação da ferramenta é analisada no Capítulo 6, que apresenta seus casos de uso e diagramas de classes. No Capítulo 7, discute-se a análise de recursos e a gestão do projeto, bem como o total de horas planejadas e gastas na execução do trabalho.

Por fim, as considerações finais são descritas no Capítulo 8, o qual apresenta uma síntese das conclusões obtidas em cada capítulo, as contribuições do trabalho e propostas de trabalhos futuros.

## 1.2 TRABALHOS RELACIONADOS

A recuperação de imagens por conteúdo é um campo de pesquisa que vem sendo muito explorado nas últimas décadas (SINGHA; K.HEMACHANDRAN, 2012). Isso se deve ao fato da já citada explosão de mídias digitais e da dificuldade associada à extração de informações que descrevam claramente o conteúdo das imagens, seja por meio de texto ou de informações relacionadas a cor, textura e forma.

O trabalho de (PENATTI, 2009) envolve a busca por imagens na Web, o qual é muito propício à explosão da web 2.0, na qual houve um aumento exponencial da quantidade de imagens compartilhadas. O trabalho fundamenta a criação de uma ferramenta eficiente para a busca de imagens por conteúdo e desenvolve um estudo comparativo de descritores por meio de duas abordagens. A primeira analisa a complexidade assintótica dos algoritmos utilizados para extração de vetores de características e das funções distância associadas. A segunda abordagem compara os descritores em experimentos práticos em quatro bases de imagens diferentes, onde são analisados os tempos para se extrair as características, calcular a distância entre descritores, requisitos de armazenamento e eficácia. Os resultados sugerem complexidade linear para descritores e funções de distância e uma degradação no desempenho dos descritores em bases de imagens grandes e heterogêneas. Isso evidencia a dependência da base de imagens e suas características para um desempenho bom dos descritores.

Em (SINGHA; K.HEMACHANDRAN, 2012), a recuperação de imagens utiliza um descritor do baseado em cor e textura chamado *Wavelet Based Color Histogram Image Retrieval*. Houve uma rápida recuperação para a base denominada WANG, que contém 1000 imagens coloridas - o que demonstra boa performance para bases com poucas imagens.

Através da utilização do descritores *Local Color Histogram*, (WANG; WANG, 2001) tenta defender a tese de que o *Global Color Histogram* perde algumas informações espaciais sobre as cores da imagem. Ele cria um grid de células superpostas de modo a aplicar a cada uma o histograma de cor, assim tentando buscar as características espaciais das cores em todas as regiões da imagem. Como resultado obteve uma boa vantagem na recuperação de imagens devido as análises locais, porém o método não é sensível a rotação e translação.

O presente trabalho considera os resultados obtidos na literatura de modo a buscar um equilíbrio entre uma base de imagens condizente com os descritores escolhidos para utilização, de modo que não haja excesso de imagens o que fará diminuir o desempenho dos descritores e nem uma base heterogênea para não tornar a complexidade do trabalho fora do escopo almejado.

## 2 RECUPERAÇÃO DE IMAGENS BASEADA EM CONTEÚDO

Como mencionado, técnicas de CBIR constituem uma alternativa à busca de imagens com base em informações textuais, que requer a anotação manual de cada imagem com as características que possam ser buscadas posteriormente (FENG; SIU; ZHANG, 2003). Além de ser subjetivo (as anotações dependem da pessoa executando a tarefa), o custo associado é elevado, podendo inclusive inviabilizar o processo. Por outro lado, se informações visuais são consideradas, o processo de busca não é afetado pela ausência de descrições textuais ou por erros inseridos nos metadados.

Um dos principais componentes de um sistema de CBIR é o descritor de imagens, o qual pode ser caracterizado por: (a) um algoritmo de extração de características das imagens, as quais são organizadas na forma de vetores e tipicamente referem-se à cor, textura e/ou forma e (ii) uma função de distância capaz de associar uma medida de similaridade. Neste contexto, os parâmetros do processo de busca (denominados *queries*) são representados por uma imagem. Com base em um conjunto pré-definido de descritores, é extraído o vetor de característica desta entrada, o qual é comparado aos vetores das imagens da base (os quais já foram previamente calculados). Tal comparação é realizada com base na função de distância associada, e permite recuperar imagens similares presentes na base.

Além da definição de conceitos básicos, necessários ao entendimento do restante do trabalho (Seção 2.1), este capítulo apresenta os descritores baseados em cor (Seção 2.2) e forma (Seção 2.3) utilizados neste trabalho. Com relação aos sistemas CBIR, as Seções 2.5 e 2.6 listam, respectivamente, seus componentes básicos e problemas típicos.

### 2.1 DESCRITORES

Formalmente, pode-se definir um descritor como uma tupla  $(\epsilon_d, \delta_d)$  (TORRES; FALCAO, 2005), em que  $\epsilon_d$  é um algoritmo ou técnica capaz de extrair uma ou mais características de uma imagem e armazená-las em um vetor que as represente. Para um par de vetores de saída obtidos a partir  $\epsilon_d$ ,  $\delta_d$  define uma função capaz de compará-los. A Figura 1 esquematiza esta



**Figura 1: O descritor de imagens**

definição.

A operação definida pelo item  $\varepsilon_d$  da tupla é descrito também na literatura como “extração de características”, enquanto o item  $\delta_d$  como “funções de distância”. As seções abaixo as descrevem com detalhes.

### 2.1.1 EXTRAÇÃO DE CARACTERÍSTICAS

A extração de características é um dos elementos-chave em sistemas CBIR, sendo que idealmente devem ser precisas e perceptualmente importantes. Elas são usualmente avaliadas em relação à precisão com que recuperam formas similares de uma base. Entretanto, é preciso levar em conta outras propriedades, tais como a eficiência computacional, a qual pode ser considerada tão importante quanto efetividade em aplicações de recuperação de imagens por conteúdo.

Basicamente, características consistem em um conjunto de valores numéricos que representam (ou sumarizam) o conteúdo de uma imagem, podendo ser extraídas a partir de diferentes informações, tais como cor, textura, forma e organização espacial (TRAN, 2003). No contexto deste trabalho, os descritores utilizados baseiam-se em informações de cor e forma, como apresentado nas Seções 2.2 e 2.3.

A cor é uma característica subjetiva, podendo ser alterada pela iluminação do ambiente ou pela textura da superfície, por exemplo. Tais fatores devem ser levados em consideração na extração automática de características para fins de recuperação de imagens por conteúdo. Como vantagens, tem-se que a cor é robusta à rotação, translação e distorção (SINGHA; K.HEMACHANDRAN, 2012). São três as abordagens de pesquisa envolvendo cores na área de CBIR (TORRES; FALCAO, 2005): definição do sistema de cores adequado para cada aplicação, definição de algoritmos de extração de características, e avaliação de medidas de similaridade.

A descrição da forma de objetos é uma tarefa essencial em reconhecimento de padrões, sendo que esta é uma característica visual da imagem muito utilizada em aplicações de classificação e recuperação de imagens. Embora o propósito dos primeiros descritores de forma



fosse a classificação, o uso destes para recuperação de imagens por conteúdo tem sido cada vez maior. Entretanto, devido a fatores como ruído, defeitos, distorções e oclusão, a representação e descrição da forma é uma tarefa não-trivial (TORRES; FALCAO, 2005). As técnicas de representação e descrição de formas podem ser geralmente classificadas em duas classes de métodos: baseados em região e baseados em contorno, dependendo se as características são extraídas do contorno ou de toda a forma (TORRES; FALCAO, 2005).

Dependo do descritor utilizado para extração de características, a comparação da similaridade entre imagens pode considerar diferentes funções de distância, como discutido a seguir.

### 2.1.2 FUNÇÕES DE DISTÂNCIA

As funções de distância,  $\delta_d$ , fazem parte da definição de um descritor e são responsáveis por comparar os vetores de características para determinar a similaridade entre duas imagens.

Uma função  $\delta_d(V_Q, V_B)$ , em que  $V_Q$  e  $V_B$  representam respectivamente o vetor de características da imagem *query* e da imagem da base de comparação, em geral é uma função de distância métrica (PENATTI, 2009) que atende as seguintes propriedades:

$$\delta_d : X \times X \rightarrow \mathbb{R}^+ \quad (1)$$

$$(V_Q, V_B) \rightarrow \delta_d(V_Q, V_B) \quad (2)$$

A Equação 1 define a função de distância de domínios iguais em  $X$  e de imagem no conjunto dos reais estritamente positivos. A Equação 2 define o sentido da função de distância, que é comparar dois vetores, chamados de  $V_Q$  e  $V_B$ .

Para ser uma distância métrica, a função  $\delta_d(V_Q, V_B)$  satisfaz, para todo  $V_B$  e  $V_Q$  (PENATTI, 2009):

1.  $\delta_d(V_Q, V_B) = 0 \iff V_Q = V_B$ , identidade
2.  $\delta_d(V_Q, V_B) = \delta_d(V_B, V_Q)$ , comutatividade
3.  $\delta_d(V_Q, V_{B1}) \leq \delta_d(V_Q, V_{B2}) + \delta_d(V_{B2}, V_{B1})$ , desigualdade triangular
4.  $\delta_d(V_Q, V_{B1}) \geq 0$ , não negatividade

Alguns trabalhos da literatura analisam o desempenho de variadas funções de distância em sistemas CBIR, como (BUGATTI; TRAINA, 2009) e (WILSON; MARTINEZ, 1997). As se-

ções abaixo apresentam as funções de distância utilizadas no trabalho. Em alguns experimentos utilizaram-se variações destas, as quais serão discutidas no Capítulo 5.

Para as funções de distância apresentadas abaixo (PENATTI, 2009), os termos  $V_B$ ,  $V_Q$  e  $N$  representam os vetores de características extraídos da imagem base, da imagem *query* e o tamanho destes vetores (obrigatoriamente o mesmo).

### 2.1.2.1 Distância L1

A distância L1, também conhecida como *City-Block*, calcula as diferenças modulares entre os elementos correspondentes de dois vetores:

$$\delta_{L1}(V_B, V_Q) = \sum_{i=1}^N |V_Q(i) - V_B(i)| \quad (4)$$

### 2.1.2.2 Distância L2

A distância L2, também chamada de distância Euclidiana, é calculada pela seguinte equação:

$$\delta_{L2}(V_B, V_Q) = \sqrt{\sum_{i=1}^N (V_Q(i) - V_B(i))^2} \quad (5)$$

### 2.1.2.3 Distância $L_\infty$

A equação da distância  $L_\infty$  é dada abaixo:

$$\delta_{L_\infty}(V_B, V_Q) = \max_{1 \leq i \leq N} |V_Q(i) - V_B(i)| \quad (6)$$

As distâncias L1, L2 e  $L_\infty$  necessitam percorrer somente uma vez os vetores de características da imagem base e da *query*, sendo em geral rapidamente calculadas.

## 2.2 DESCRITORES DE COR

O olho humano é bastante sensível a cor, permitindo a distinção de objetos utilizando-a como base. Isto torna conveniente utilizar descritores que utilizam tal característica. Cada pixel

possui, geralmente, três canais de cor (correspondentes aos tons de vermelho, verde e azul, no caso do sistema de cores RGB), sendo representado por um vetor tridimensional.

O método mais direto para descrever a cor de uma imagem é através de uma estimativa probabilística da densidade, representada por um histograma. Outros métodos incluem o uso de cores dominantes, momentos de cor e diferentes espaços de cor (TRAN, 2003).

Este trabalho considera dois descritores: o *Global Color Histogram* (GCH) e o *Dominant Color Descriptor* (DCD), descritos nas Seções 2.2.1 e 2.2.2, respectivamente.

### 2.2.1 GLOBAL COLOR HISTOGRAM - GCH

O histograma de cor de uma imagem pode ser definido como um vetor onde cada entrada contém a quantidade de *pixels* de uma cor particular presentes naquela imagem. Formalmente, para um histograma denotado como  $H$ , tem-se que:

$$H = \{H[0], H[1], H[2], \dots, H[i], \dots, H[N]\}, \quad (7)$$

em que  $i$  denota a posição que representa uma cor específica e  $H[i]$  a  $i$ -ésima posição do vetor.

Para possibilitar a comparação de imagens de diferentes tamanhos, os histogramas de cor devem ser normalizados como segue:

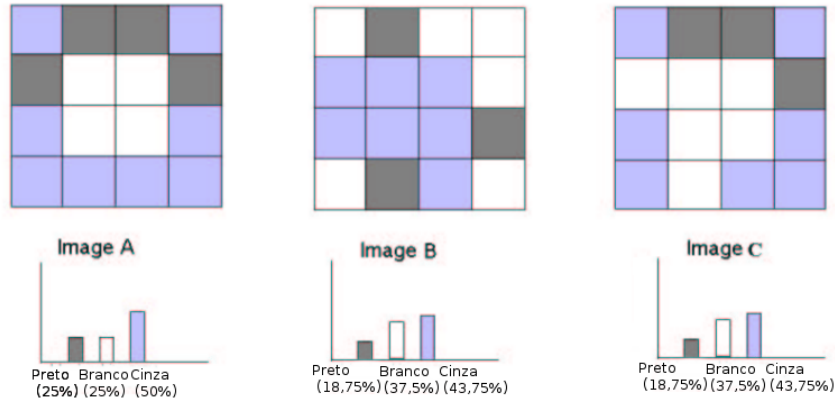
$$H' = \{H'[0], H'[1], H'[2], \dots, H'[i], \dots, H'[N]\}, \quad (9)$$

em que  $H'[i] = \frac{H[i]}{T}$ , sendo  $T$  é o total de pixels da imagem.

O descritor GCH nada mais é do que o histograma de cor calculado a partir das características globais de cor em uma imagem. Ele é muito utilizado em diversos sistemas CBIR (SHARMA; RAWAT, 2012). A representação do espaço RGB em um histograma global de cor contendo  $N$  entradas faz com que uma entrada qualquer  $i$  agregue um conjunto de cores composto por um sub-cubo do espaço. Quanto menor o valor de  $N$ , maior a chance de cores diferentes se localizarem no mesmo sub-cubo. Tipicamente os estudos na literatura usam  $N = 64$  (WANG; WANG, 2001).

A similaridade entre duas imagens é determinada pela distância entre seus histogramas de cor, a qual pode ser calculada de diferentes formas, incluindo por meio das funções apresentadas

na Seção 2.1.2. Para ilustrar a aplicação do descritor GCH, Wang e Wang (2001) apresenta um exemplo, mostrado na Figura 2.



**Figura 2: Exemplo de histograma global de cor para três imagens**

**Fonte: Adaptado de (WANG; WANG, 2001)**

Foram utilizados histogramas com 3 entradas,  $N = 3$ , correspondendo às cores preta, branca e cinza. Os histogramas normalizados para as imagens A, B e C são, respectivamente:  $H'_A = \{25\%, 25\%, 50\%\}$ ,  $H'_B = \{18.75\%, 37.5\%, 43.75\%\}$  e  $H'_C = \{18.75\%, 37.5\%, 43.75\%\}$ . Percebe-se que embora, B e C, sejam visualmente diferentes, apresentam os mesmos valores em seus histogramas de cor.

Supondo que a distância L2 seja utilizada para calcular a diferença entre A e B, temos:

$$d_{GCH}(A, B) = \sqrt{(0.25 - 0.1875)^2 + (0.25 - 0.375)^2 + (0.5 - 0.4375)^2} = 0.153 \quad (11)$$

A distância L2 aplicada nos descritores das imagens B e C resulta em zero. Isso evidencia um dos problemas do descritor GCH: ele não inclui informações a respeito da distribuição de cor das imagens, permitindo entradas distintas visualmente sejam classificadas como próximas.

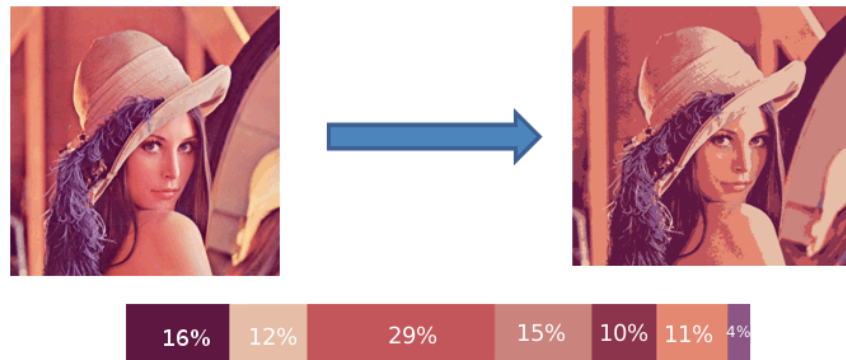
## 2.2.2 DOMINANT COLOR DESCRIPTOR

Ao contrário do GCH, o DCD preocupa-se apenas com a recuperação de cores denominadas “mais representativas” da imagem, a partir de um determinado critério (MAN, 2004). Este descritor é definido como (OHM et al., 2001):

$$F = \{\{C_i, P_i, V_i, \}_s\}, i = 1, 2, \dots, N \quad (13)$$

em que  $N$  representa o total de cores dominantes encontradas,  $C_i$  o valor da cor em um espaço de cores qualquer (como, por exemplo, RGB),  $P_i$  o percentual da cor  $C_i$  na imagem,  $V_i$  a variância

da cor na região e  $s$  a coerência espacial que representa a homogeneidade da cor dominante na imagem. A Figura 3 ilustra a aplicação do DCD em uma imagem.



**Figura 3: Descritor de cor dominante.**

**Fonte: Adaptado de (BITTON, 2013)**

O procedimento de extração das cores mais representativas usa o Algoritmo Generalizado de Lloyd (OHM et al., 2001) para agrupar os *pixels*. O procedimento inicia-se com apenas um agrupamento de cores e então divide-se em mais centróides até atingir um critério de parada (número de iterações, por exemplo). O algoritmo calcula então o percentual de cada cor encontrada e os valores de variância e coerência espacial.

O DCD é um dos descritores presentes no padrão MPEG-7, o qual define um conjunto de descritores para execução em sistemas CBIR (OHM et al., 2001).

### 2.3 DESCRITORES DE FORMA

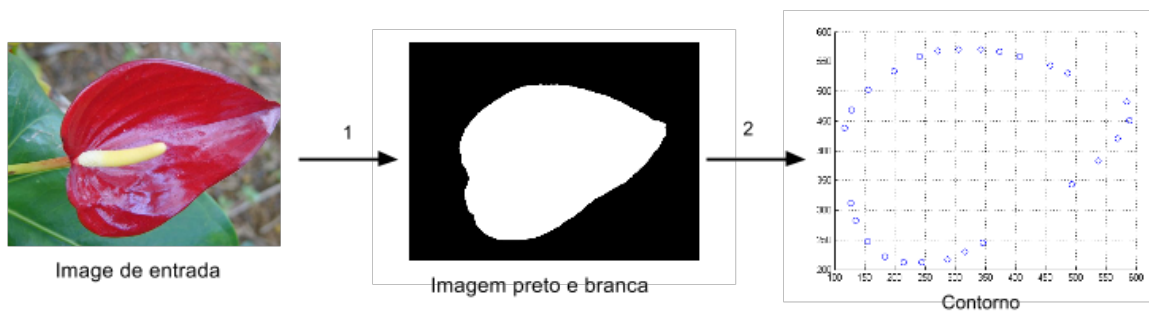
Definir a forma de um objeto em uma imagem é geralmente uma tarefa difícil. O processamento computacional associado demanda a descrição de formatos complicados e, embora existam diversos métodos, nenhum é totalmente aceito (TRAN, 2003). A visão humana tolera diferentes distorções e fenômenos relacionados à forma, o que requer que descritores deste tipo sejam robustos a objetos rotacionados, transladados, escalonados, com ruído, distorcidos ou parcialmente encobertos (PENATTI, 2009 apud ZHANG; LU, 2004).

Zhang e Lu (2004) apresentam uma taxonomia de descritores que divide-os em dois grupos: baseados em contorno ou região. No primeiro caso, extraem informações relativas somente à fronteira externa de um objeto. Exemplos incluem os descritores de Fourier e representações geométricas (curvatura e espessura da fronteira, por exemplo) (TRAN, 2003). O Descritor de Feixo Convexo (DFC), utilizado neste trabalho, se encaixa nesta categoria.

Por outro lado, os descritores baseados em região extraem e comparam toda a região do objeto. Segundo Penatti (2009), esta abordagem é mais robusta do que a anterior, por considerar todo o objeto e não apenas a sua fronteira. Um exemplo de descritor baseado em região utilizado neste trabalho é o *Scale Invariant Feature Transform* (SIFT), descrito na Seção 2.3.2.

### 2.3.1 DESCRITOR DE FEIXO CONVEXO

Feixo convexo é definido como o menor conjunto de pontos que contém toda a forma a ser capturada na imagem (BERG et al., 2000). Ele fornece os pontos que contornam a imagem em uma distribuição espacial relacionada à forma nela contida. A Figura 4 ilustra as duas primeiras etapas para extração do vetor de características relacionado ao descritor DFC, sendo elas a segmentação (no contexto deste trabalho, representada pela transformação da imagem de entrada em outra contendo apenas as cores branco e preto, representando objeto e fundo, respectivamente) e a obtenção dos pontos de feixo convexo que contornam a imagem.

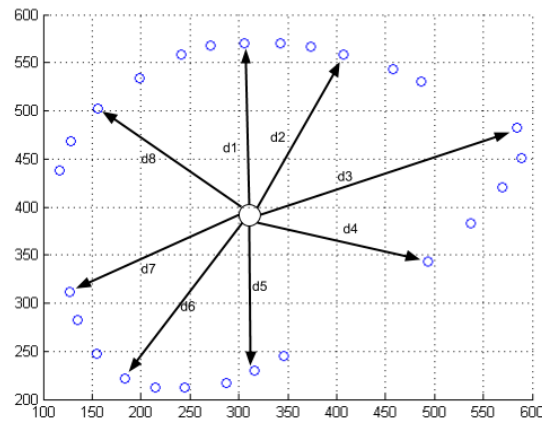


**Figura 4: Descritor de Feixo Convexo - As duas primeiras etapas do DFC**

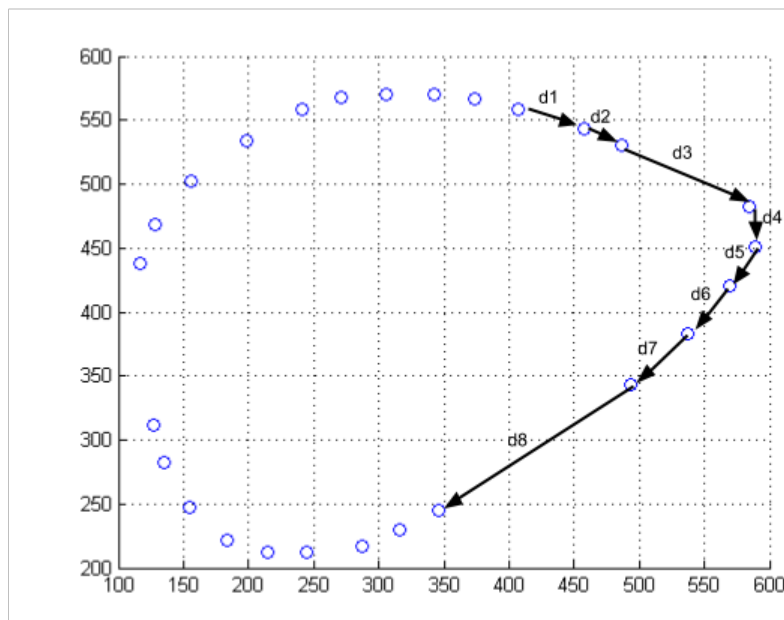
A próxima etapa extrai o desvio padrão das distâncias do centro para todos os pontos no contorno. A Figura 5 ilustra as distâncias para oito pontos da região delimitada.

A última etapa extrai o desvio padrão das distâncias entre pontos sequenciais do contorno obtido com o feixo convexo, conforme ilustra a Figura 6.

O descritor possui, portanto, três dimensões, sendo a primeira o total de pontos no contorno, a segunda o desvio padrão das distâncias do centro aos pontos do contorno e a terceira o desvio padrão das distâncias entre pontos sequenciais. Com essas informações, pode-se obter uma caracterização aproximada da forma da imagem a ser descrita que permita a análise de similaridade com outras imagens.



**Figura 5: DFC - Extração das distâncias entre o centro e os pontos de contorno**



**Figura 6: DFC - Desvio padrão entre pontos sequenciais do contorno**

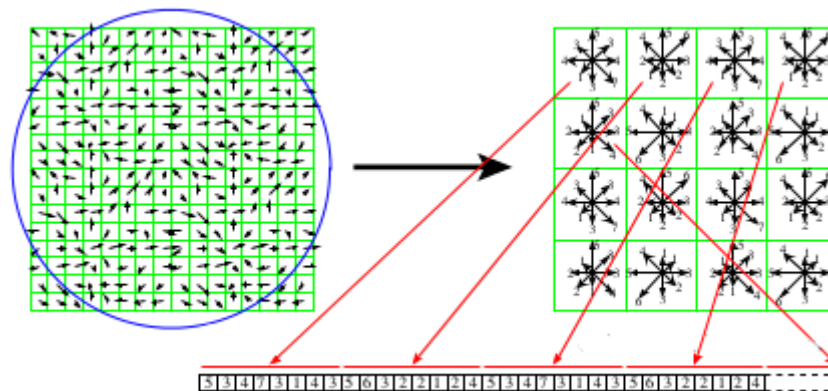
### 2.3.2 SCALE-INVARIANT FEATURE TRANSFORM (SIFT)

A técnica SIFT é amplamente aplicada na detecção de objetos, reconhecimento de panoramas e reconstrução tridimensional. Isso se dá através da seleção de muitos parâmetros que afetam a eficiência da recuperação de imagens (JR; TORRES; GOLDENSTEIN, 2009).

O SIFT possui quatro etapas para a construção do descritor da imagem sendo elas: detecção de espaços em escala, localização de pontos-chave, atribuição de orientações e descrição dos pontos-chave. A detecção de espaços em escala tem como função buscar por regiões na imagem que podem ser encontradas mesmo em diferentes pontos de vista.

A localização de pontos-chave é centrada em encontrar pontos relevantes na imagem que não mudam com a orientação. A atribuição de orientações diz respeito à orientação dos pontos-chave encontrados e por fim a descrição dos pontos-chave cria o descritor com base em todas as informações e pontos encontrados (JR; TORRES; GOLDENSTEIN, 2009).

A Figura 7 ilustra os quatro passos realizados pelo SIFT. Ocorre à esquerda a detecção dos espaços em escala, em cada espaço se detecta um ponto-chave e a orientação do mesmo é atribuída. Já à direita se encontra o descritor que condensa em cada quadrado as diversas orientações de uma região da imagem à esquerda.



**Figura 7: SIFT - Ilustração do processo de extração das características (JR; TORRES; GOLDENSTEIN, 2009).**

## 2.4 CLASSIFICAÇÃO

Segundo (SILVA, 2007), há dois tipos de classificação: não-supervisionada e supervisionada. Na primeira, não há definição prévia nem da quantidade de classes existentes e nem a rotulação das amostras de treinamento. Portanto, o algoritmo classificador é responsável por agregar imagens em classes geradas em tempo de execução. A técnica denominada *clustering* pertence a esta categoria.

Por outro lado, nas técnicas de classificação supervisionada são fornecidas amostras de treinamento indicando-se diretamente a qual classe pertencem, de forma que possa-se inferir a classe de novas entradas comparando-as com o conjunto de treinamento. Este processo inicia com a etapa de treinamento do classificador, dado que há conhecimento prévio sobre as informações agregadas nas classes. Em seguida, ocorre a classificação das classes subsequentes baseado nos parâmetros de treinamento (LIU, 2005).

Neste trabalho, será adotada a classificação supervisionada. O classificador utilizado nos

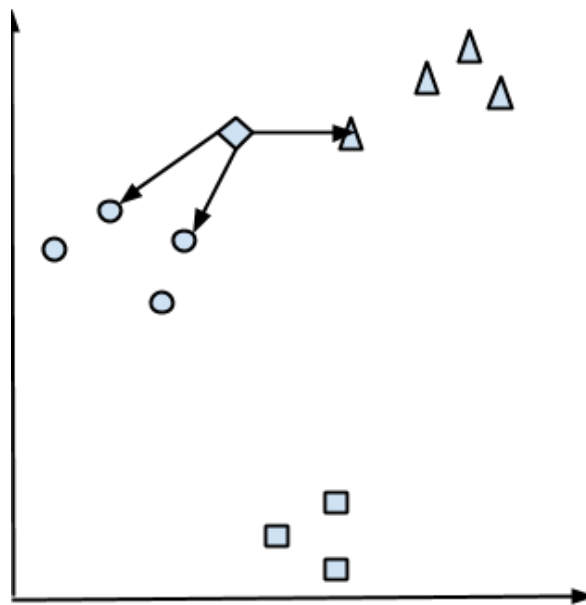


experimentos futuros é o *K-Nearest Neighbors* (KNN), sendo descrito na próxima seção.

#### 2.4.1 K-NEAREST NEIGHBORS (KNN)

O classificador KNN é do tipo supervisionado e, conforme (BEYER et al., 1999), é utilizado para classificar amostras desconhecidas. Isso é feito analisando-se à qual classes pertencem as  $k$  amostras mais similares do espaço de busca. A amostra sendo classificada é considerada pertencente à classe que contem mais amostras próximas.

A Figura 8 ilustra um exemplo em que há um elemento desconhecido representado por um losango e o  $k=3$ , ou seja, ele irá ser classificado baseado na análise da(s) classe(s) dos três vizinhos mais próximos.



**Figura 8: Exemplo de classificação por KNN**

Observe que o losango está próximo de dois elementos da classe representada por um círculo e de um elemento da classe representada por um triângulo. Portanto, por votação, o elemento desconhecido será classificado como participante da classe representada pelo círculo.

Cada elemento a ser classificado no KNN tem atributos que o definem, os quais são utilizados para se calcular as  $k$  amostras mais similares. No contexto deste trabalho, tais atributos são representados pelos vetores de características. O KNN pode ser utilizado mediante os passos descritos abaixo por (LIMA, 2008):

1. dado um conjunto de exemplos de treinamento;
2. definir uma métrica para calcular a distância entre os exemplos de treinamento;

3. definir o valor de K (o número de vizinhos mais próximos que serão considerados pelo algoritmo);
4. calcular a distância entre o exemplo desconhecido e o outros exemplos do conjunto de treinamento;
5. identificar os K vizinhos mais próximos;
6. utilizar o rótulo da classe dos vizinhos mais próximos para determinar o rótulo de classe do exemplo desconhecido.

Neste trabalho, o KNN será utilizado na etapa de validação de descritores e para combinar a recuperação de imagens com ontologias.

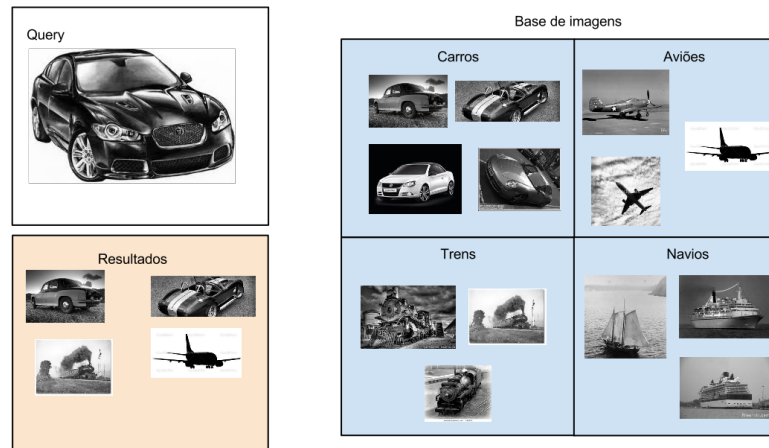
#### 2.4.2 MEDIDORES DE DESEMPENHO DA CLASSIFICAÇÃO

Na sequência, serão discutidos as métricas utilizadas para análise dos resultados de classificação.

##### 2.4.2.1 Precisão e revocação

As buscas em um problema de classificação são um problema de decisão binária: ou o elemento pertence à classe do objeto da imagem de busca ou não. Em um problema deste tipo, o classificador indica se as saídas são positivas ou negativas. Tais decisões podem ser representadas em uma estrutura conhecida como “tabela de contingência”. Esta tabela é dividida em quatro células:

1. Positivo verdadeiro (PV): conjunto de respostas da busca que pertencem ao conjunto da imagem de busca (conjunto solução);
2. Falso positivo (FP): conjunto de respostas que não pertencem ao conjunto da *query* e aparecem na resposta;
3. Negativo verdadeiro (NV) : exemplos que não pertencem à classe da *query* e também não aparecem na resposta;
4. Falso Negativo (FN): exemplos que pertencem à classe da *query* mas não aparecem na resposta (DAVIS; GOADRICH, 2006).



**Figura 9:** Exemplo de busca de imagens ilustrando a tabela de contingência

	Positivo	Negativo
Verdadeiro	2	7
Falso	2	2

**Tabela 1:** Exemplo de tabela de contingência para a *query* da Figura 9.

Para a *query* da Figura 9, apresenta-se a sua tabela de contingência (Tabela 1).

A partir da tabela de contingência, definem-se dois indicadores de desempenho de uma busca, denominados precisão e revocação. A precisão, denotada por  $P$ , pode ser definida pela seguinte relação:

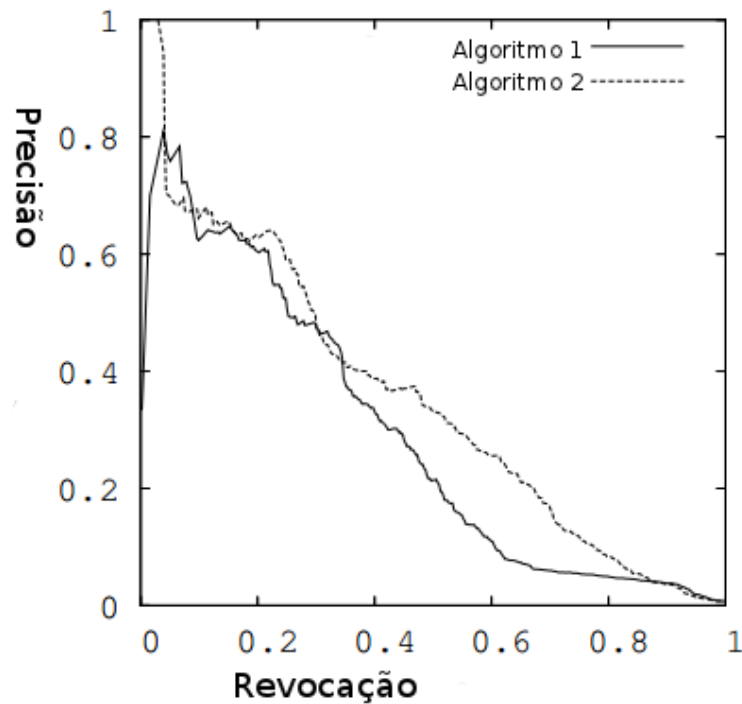
$$P = \frac{PV}{PV + FP} \quad (15)$$

que indica a quantidade de positivos verdadeiros recuperados em relação ao montante que pode conter falsos positivos. Seu valor ideal é igual a 1, quando  $FP = 0$  (todos os elementos do conjunto da entrada da busca foram encontrados). A precisão avalia os acertos do conjunto, mas não os erros. Este é o papel da revocação.

A revocação (denotada por  $R$ ) pode ser definida pela relação:

$$R = \frac{PV}{PV + FN} \quad (17)$$

que indica a quantidade de positivos verdadeiros recuperados em relação ao montante que pode conter falsos negativos. Seu valor ideal também é igual a 1, já que neste caso  $FN = 0$ , e a



**Figura 10: Exemplo de curva precisão e revocação**

**Fonte: Adaptado de Davis e Goadrich (2006)**

imagem não apresenta falsos negativos.

Percebe-se a complementaridade entre a precisão e a revocação. Isoladamente, cada uma dessas medidas não é um bom indicador de desempenho. Uma *query* pode apresentar alta precisão (bom número de positivos verdadeiros) e baixa revocação (grande quantidade de falsos negativos e vice-versa).

Para experimentos que apresentam várias entradas de teste, é possível apresentar os dados sob a forma de curvas de precisão e revocação. Elas são construídas sob a forma de gráfico onde o eixo das abcissas apresenta a revocação e o eixo das ordenadas a precisão. Cada ponto corresponde aos valores calculados de precisão/revocação para uma entrada de testes. A Figura 10 apresenta um exemplo de curva precisão e revocação para análise de desempenho de dois algoritmos de detecção de câncer de mama. Segundo Davis e Goadrich (2006) estas curvas evidenciam que ambos os algoritmos de classificação precisam de melhorias.

Os conceitos de precisão, revocação e curva precisão e revocação serão utilizados no Capítulo 5 ao longo de suas três etapas.

#### 2.4.2.2 Matriz de confusão

A precisão e revocação avaliam apenas informações referentes à própria classe do objeto da busca. Uma forma mais detalhada de avaliar os erros de classificação é fornecida através da matriz de confusão, a qual resume a performance de um classificador com respeito a algumas entradas de teste. Ela consiste em uma matriz bidimensional, em que uma dimensão indica qual a classe verdadeira de um conjunto de entradas de teste e a outra dimensão indica qual a classe atribuída pelo classificador (TING, 2010).

A Tabela 2 apresenta um exemplo de matriz de confusão. Neste exemplo, foram realizadas buscas para 12 entradas entre as classes “Antúrio”, “Copo de Leite” e “Rosa”. O somatório dos valores ao longo da linha indica o total de entradas reais para cada classe. As colunas indicam a classe encontrada pelo classificador.

Dentre as entradas de teste no exemplo, quatro delas eram da classe “Antúrio”, sendo todas classificadas como “Antúrio”. Para a classe “Copo de Leite”, composta também de quatro entradas de teste, duas entradas foram classificadas como sendo “Antúrio” e duas como “Copo de Leite”. Para a classe rosa, três entradas foram classificadas como “Antúrio” e uma como “Copo de Leite”.

Real / Escolhida	Antúrio	Copo de Leite	Rosa
Antúrio	4	0	0
Copo de leite	2	2	0
Rosa	3	1	0

**Tabela 2:** Exemplo de matriz de confusão

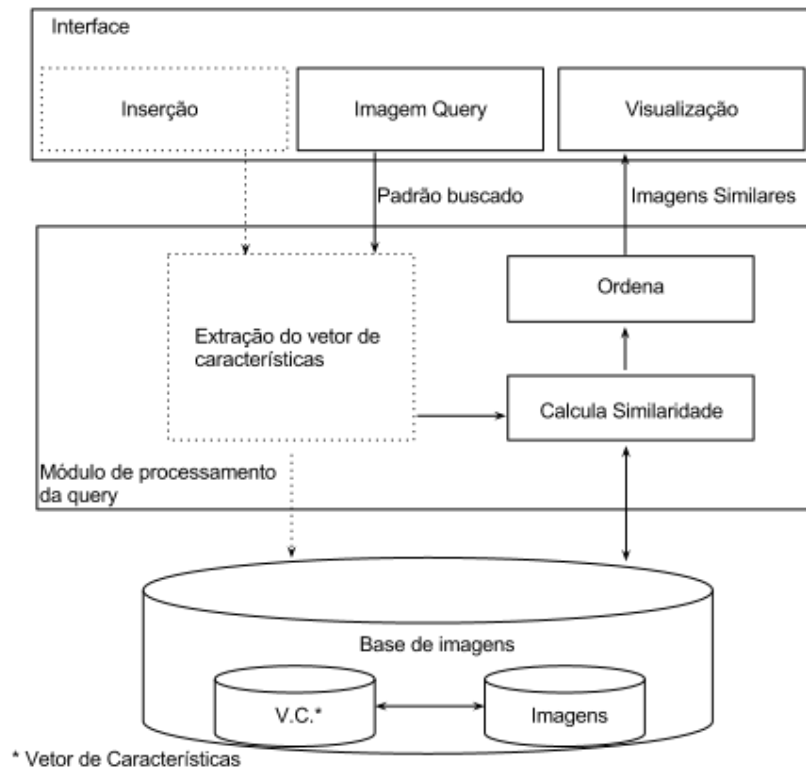
A notação da Tabela 2 será utilizada no restante deste trabalho. A matriz de confusão será utilizada na etapa de validação de descritores (Seção 5.2) para indicar erros de classificação nos experimentos.

## 2.5 COMPONENTES BÁSICOS DE UM SISTEMA DE RECUPERAÇÃO DE IMAGENS POR CONTEÚDO

Os sistemas de CBIR tipicamente possuem uma arquitetura composta pelos componentes ilustrados na Figura 11, baseada em (TORRES; FALCAO, 2005):

As próximas seções descrevem tais componentes básicos:

- interface, onde o usuário insere uma imagem de entrada e visualiza aquelas com determinado grau de similaridade (Seção 2.5.1);



**Figura 11: Arquitetura típica de um sistema CBIR.**

**Fonte: (TORRES; FALCAO, 2005)**

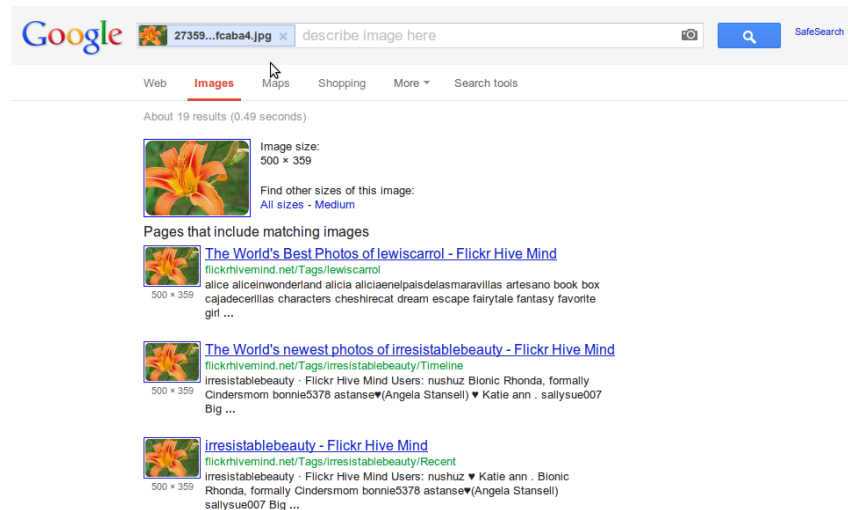
- módulo de processamento, composto por descritores usados para analisar as características utilizadas para comparação de similaridade e ordenação dos resultados (Seção 2.5.2);
- base de imagens, contendo o “espaço de busca” por imagens similares (Seção 2.5.3).

### 2.5.1 INTERFACE

A interface de um sistema CBIR é o componente diretamente ligado ao usuário. Geralmente, ela é disponibilizada na *web* e apresenta um campo de entrada, onde insere-se a imagem, e um espaço para visualizar os resultados.

Como os sistemas de busca de imagem são utilizados pelo público em geral, as interfaces visam ser o mais simples quanto possível, abstraindo detalhes técnicos e operacionais da busca. Como exemplo, tem-se o buscador de imagens do *Google*, ilustrado na Figura 12.

Em trabalhos de pesquisa, a interface de sistemas de CBIR tende a apresentar mais detalhes de como as *queries* são processadas, dado que o procedimento de busca é continuamente modificado, sendo necessário visualizar com facilidade o impacto destas mudanças. Um exemplo de sistema deste tipo é o EVA, desenvolvido para estudos de desempenho de descritores reali-



**Figura 12: Interface de buscas por imagem do google em Janeiro de 2013.**

zados na dissertação de (PENATTI, 2009). O EVA trabalha com a noção de “experimento” que trata-se de uma *query* realizada com um dado grupo de descritores. É possível escolher quais descritores usar, além de informações como o tempo de duração do experimento e para cada descritor além da visualização dos resultados.

No Capítulo 4, será discutida a interface utilizada para visualização dos resultados para os experimentos realizados ao longo do trabalho.

## 2.5.2 MÓDULO DE PROCESSAMENTO

O módulo de processamento da *query* é o componente mais complexo de um sistema de CBIR, pois envolve todo o processo entre a solicitação do usuário e a visualização dos resultados. Basicamente, este módulo extrai um vetor de características da imagem dada como entrada pelo usuário e aplica a função de distância associada (pelo descritor) para avaliar sua similaridade em relação às demais imagens da base de dados. Aquelas que possuem maior índice de semelhança com a imagem de entrada são enviadas ao módulo de interface (visualização).

## 2.5.3 BASE DE IMAGENS

A base de imagens define o espaço de busca do sistema. Em sistemas de busca na *web*, as bases são comumente compostas de um número muito grande de arquivos, o que tornaria impraticável armazená-los em um só local. No intuito de contornar tais problemas, estes sistemas indexam as referências para os arquivos originais e pequenas miniaturas, utilizadas para agilizar a exibição dos resultados.

Para estudos em CBIR, as bases tendem a ser mais restritivas em número e variedade de objetos, visando facilitar a análise dos resultados e minimizar preocupações de manutenção com sistemas de grande porte. É o caso deste trabalho, que será realizado utilizando-se da base “102 Category flower dataset” (NILSBACK; ZISSERMAN, 2012), a qual é composta por imagens de 102 espécies de flores, grande parte delas obtidas na *web* - ela será discutida no Capítulo 4.

## 2.6 PROBLEMAS DE UM SISTEMA CBIR

Os problemas associados com CBIR envolvem baixo desempenho de descritores (devido a extensão da base de imagem), problemas de precisão associados à heterogeneidade do espaço de busca do sistema e, por fim, a implementação dos algoritmos que descrevem os descritores e seus vetores de características (que são determinantes para a eficiência da busca).

Dois problemas também característicos de sistemas CBIR são o *semantic gap* e o *sensorial gap*, que estão relacionados à interpretação humana das imagens à representação dada pelos descritores e algoritmos de extração de informação. Tais problemas serão explicados na próxima seção.

### 2.6.1 SEMANTIC E SENSORIAL GAP

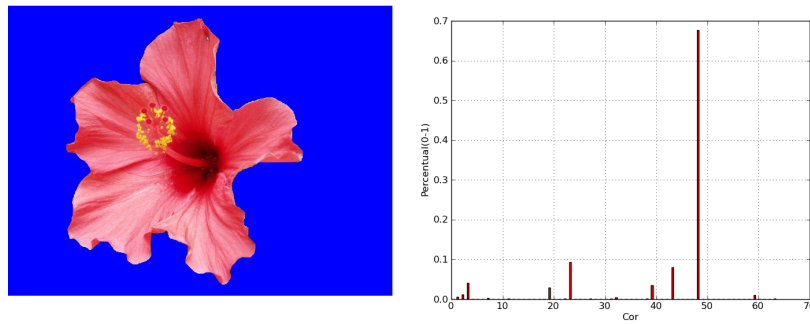
Para Smeulders et al. (2000) o *semantic* e o *sensorial gap* são os dois problemas críticos em um sistema CBIR, evidenciando a dificuldade em correlacionar os descritores com atributos em alto nível que o usuário compreende e utilizará para buscar as imagens.

Os vetores de características, extraídos a partir dos sistemas CBIR, em geral não são intuitivos para o usuário comum. A falta de relação entre as informações extraídas a partir do descritor e as informações obtidas a partir da imagem pelo usuário chama-se *semantic gap*, ou seja, há uma lacuna entre o significado de alto nível da imagem - dado pelos seres humanos - e as características de baixo nível extraídas pelos algoritmos, que são geralmente números representando um atributo da imagem.

Para exemplificar o *semantic gap*, considere a Figura 13. Ela é composta de uma imagem e o seu respectivo histograma global de cor (GCH) com 64 valores. Surge um pergunta: esta representação do descritor é intuitiva para o usuário final? A diferença de representações entre o que o usuário espera e a resposta do sistema é o resultado do *semantic gap*.

Outro problema com relação às técnicas de CBIR diz respeito a sua incapacidade de relacionar a representação de uma imagem com um objeto do mundo real, ou seja, não há interpreta-





**Figura 13: *Semantic gap* : Uma imagem e o seu histograma global de cor obtido pelo descritor GCH**

ção física por parte do sistema, acarretando em suposições errôneas como segue com a seguinte questão: qual das meninas da Figura 14 possui a de menor altura?



**Figura 14: *Sensorial Gap* - um dos problemas das técnicas de CBIR**

**Fonte: Nevit Dilmen**

No senso comum, a resposta é simples, pois temos uma representação mental do que vem a ser a perspectiva em imagens. Entretanto, um sistema CBIR não conta com tal atributo sem um extenso trabalho para combater essa lacuna entre representação humana e de máquina, denominada *sensorial gap*.

Ao considerar o *sensorial gap*, a menina mais à direita seria associada à maior altura e a mais à esquerda seria a menor. Percebe-se que as técnicas de CBIR não são suficientes para resolver o problema, sendo necessária a aplicação de uma técnica complementar.

Clara é a importância em se minimizar tanto o *semantic* quanto *sensorial gap* de modo a não conduzir erros de representação mesmo com um sistema formado por descritores eficientes. Na literatura, encontra-se alguns sistemas que visam combater tais *gaps*. A proposta de um

sistema de recuperação de imagens baseado em ontologias é uma das maneiras de fazê-lo, e será explorada neste trabalho. Em (MARQUES; MAYRON; BORBA, 2006) um estudo a respeito de como incorporar a visão humana nos sistemas de reconhecimento de imagem trata da questão do *semantic gap*.

## 2.7 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo, a fundamentação teórica foi descrita de forma transparente, de forma a evitar inconsistências entre o modelo teórico e o modelo implementado via *software*. Os descritores foram abordados juntamente com a função de distância que deve medir a similaridade entre imagens e seus respectivos vetores de características.

A base de imagem também foi descrita e os problemas de CBIR que podem estar associados à baixa performance dos descritores também foram levantados, de modo a fundamentar resultados. As ferramentas de manipulação dos descritores - como por exemplo função de ordenação - foram explicadas como forma de complementar toda a parte teórica do projeto de software para realizar o experimento.

Por fim, foram apresentadas as fundamentações teóricas das duas abordagens a serem comparadas e os métodos que serão utilizados para realizar o experimento citado, assim permitindo uma correta análise de ambos os métodos e uma conclusão plausível sobre a eficiência de um sobre o outro.

### 3 ONTOLOGIAS

O objetivo deste capítulo é apresentar conceitos relativos à ontologias e como podem ser empregadas juntamente com as técnicas de CBIR para recuperação de imagens.

A definição de ontologias e conceitos relacionados são apresentados na Seção 3.1, e sua construção na Seção 3.2. A Seção 3.3 relaciona as ontologias e a recuperação de imagens e como a literatura aborda ambos os temas.

Seguindo o modelo do capítulo anterior, a seção 5.4 apresenta os componentes típicos de um sistema de recuperação de imagens baseado em ontologias tal como abordado neste trabalho.

#### 3.1 ONTOLOGIAS

A palavra Ontologia vem do grego *ontos* (ser) e *logos* (palavra). Na filosofia, a ontologia é a “ciência do que é”. Como disciplina, ela tem como objetivo criar sistemas de classificação para facilitar a organização da realidade (OSHIRO, 2008). No âmbito da *Web Semântica*, a definição mais utilizada para ontologia é proposta por Gruber (1993), sendo ela “a especificação de uma conceitualização”. Entende-se por conceitualização uma visão abstrata e simplificada do mundo a qual deseja-se representar.

Cada base ou sistema de conhecimento ou agente está comprometido com algum tipo de conceitualização, seja ele implícito ou explícito. As ontologias são explícitas e formais, pois seus elementos e restrições são claramente definidos, contendo axiomas que restringem a interpretação e uso daqueles.

Para a inteligência artificial, a ontologia está mais ligada à modelagem de conhecimento. Considera-se ela como um artefato, com vocabulário estruturado em uma taxonomia, suas definições e um conjunto de axiomas (FREDDO, 2010 apud Guizzardi, 2005).

O consórcio W3C define a ontologia como “a definição dos termos utilizados na representação de uma área de conhecimento”. Este órgão especifica que as ontologias devem apresentar descrições para os seguintes tipos de conceito (OSHIRO, 2008 apud W3C, 2013):

- Classes ou “coisas”.
- Relacionamentos entre as “coisas”.
- Propriedades ou atributos que as “coisas devam possuir”.

Para este trabalho, adota-se como definição de ontologia esta última, sendo ela empregada como artefato de apoio na recuperação de imagens. Na próxima Seção discute-se o processo de criação de uma ontologia, o qual será utilizado no Capítulo 5 quando a ontologia de domínio for de fato construída.

### 3.2 CONSTRUÇÃO DE ONTOLOGIAS

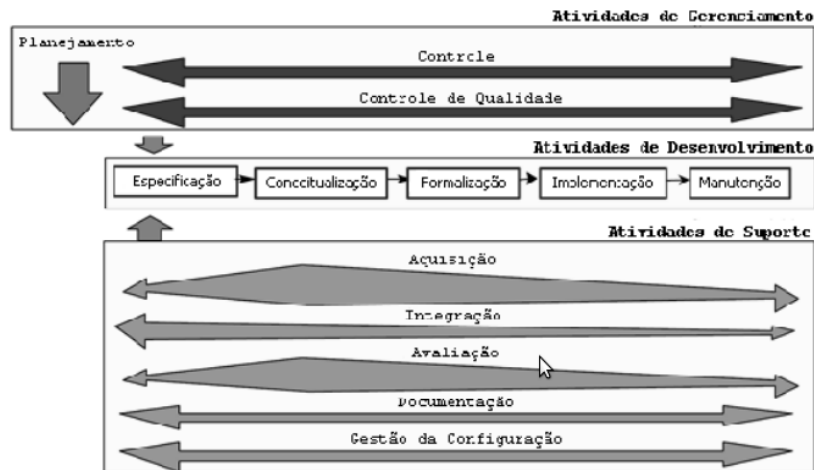
Existem diversas metodologias para o desenvolvimento de ontologias. Em Freddo (2010) são apresentadas algumas delas. Neste trabalho, a metodologia utilizada para construção de ontologias é a *Methontology* (CORCHO et al., 2005).

O *Methontology* foi uma das metodologias precursoras para a construção de ontologias. Foi criado no laboratório de Inteligência Artificial da Politécnica de Madrid em 1996. Propõe uma série de atividades categorizadas em **gerenciamento**, **suporte** e **desenvolvimento**. Este último é executado em sequência, enquanto o suporte e gerenciamento são executados concomitantemente às atividades de desenvolvimento (FREDDO, 2010).

A figura 15 ilustra as atividades do *methontology* e suas respectivas categorias. Neste trabalho discutem-se apenas as atividades de desenvolvimento, empregadas na construção de ontologia realizada no capítulo 5.

O processo de desenvolvimento de ontologias segundo o *Methontology* é sequencial e iterativo. Ele define um conjunto de atividades necessárias para sua construção, sendo elas (FREDDO, 2010 apud CORCHO et al., 2005):

- **Especificação:** Define o escopo e os objetivos da ontologia. Questiona os porquês da ontologia ser construída e sua utilização. As resposta da etapa constituem um documento de especificação de requisitos da ontologia.
- **Conceitualização:** Captura e estrutura o conhecimento do domínio em um modelo conceitual.
- **Formalização:** Transforma o modelo conceitual em um modelo computável por máquina, escrito em uma linguagem de especificação, tipicamente, em lógica de primeira ordem.



**Figura 15: Atividades do methontology**

Fonte: (FREDDO, 2010 apud CORCHO et al., 2005)

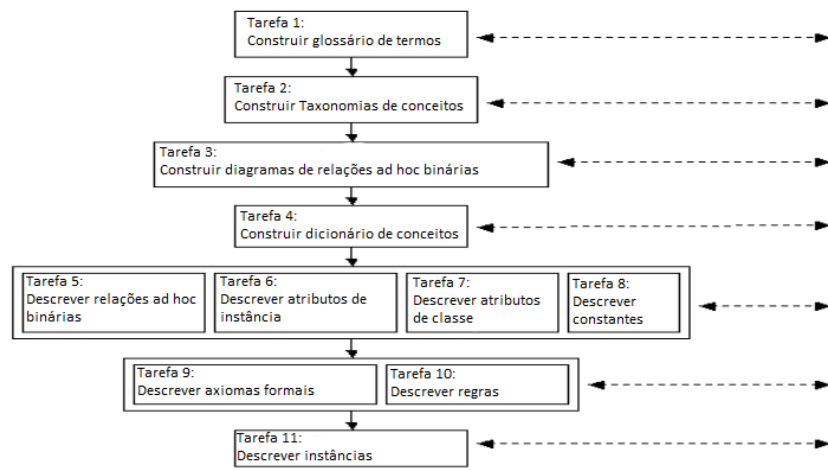
- **Implementação:** Torna a ontologia computável pela tradução em linguagem de ontologias (Ex: OWL).
- **Manutenção:** Atualizações e correções na ontologia.

A atividade de maior complexidade é a conceitualização, sendo ela descrita na Seção abaixo.

### 3.2.1 CONCEITUALIZAÇÃO NO METHONTOLOGY

A conceitualização no *Methontology* permite a modelagem dos seguintes componentes (CORCHO et al., 2005):

- **Conceitos:** As chamadas “coisas”, sendo em geral apresentadas sob a forma de taxonomia.
- **Relações:** Representam algum tipo de associação entre conceitos em um domínio. Uma importante relação chama-se “Subclasse de”, utilizada para construir uma taxonomia de classes. Cada relação possui em geral uma relação inversa, como as relações *vende a* e *compra de*, por exemplo.
- **Indivíduos:** São usados para representar elementos ou indivíduos em uma ontologia. Um exemplo de instância de *país* pode ser *Brasil* ou *França*.
- **Constantes:** São valores numéricos que não se modificam durante o tempo.



**Figura 16: Componentes do methontology**

Fonte: Adptado de Corcho et al. (2005)

- **Atributos:** Descrevem propriedades de instâncias e conceitos. Distinguem-se em dois grupos: Atributos de instância e de classe. Os primeiros referem-se a indivíduos, como por exemplo o atributo “nome” é relativo a um país. Os últimos referem-se as classes e são herdados por subclasses, mas nunca por indivíduos. Um exemplo refere-se por exemplo é o atributo “controle” em uma companhia, que pode ser “público” ou “privado”.
- **Axiomas formais:** São expressões lógicas que são sempre verdadeiras e são usadas em geral para especificar restrições em uma ontologia. Por exemplo: Um país só pode pertencer a um continente.
- **Regras:** São usadas em geral para inferir conhecimento de uma ontologia, como valores de atributos, relações entre instâncias, etc.

Dados os componentes acima, os ontologistas podem construir uma ontologia. Este processo é dividido em 11 tarefas e é ilustrado na figura 16.

Os passos para o desenvolvimento segundo esta metodologia são:

1. **Construir glossário de termos:** O ontologista constrói um glossário de termos que inclui aqueles relevantes para o domínio (conceitos, instâncias, atributos, relações, etc), as suas descrições em linguagem natural, seus sinônimos e acrônimos.
2. **Construir taxonomias de conceitos:** Quando o glossário atingir um número considerável de termos, o ontologista constrói uma hierarquia de conceitos.

3. **Construir diagramas de relações *ad hoc* binárias:** Uma vez que a taxonomia foi construída esta etapa visa estabelecer relações *ad hoc* entre conceitos de mesma taxonomia e taxonomia distinta.
4. **Construir dicionário de conceitos:** Uma vez que as taxonomias de conceito e diagramas de relações binárias *ad hoc* tenham sido criados, o ontologista deve especificar quais são as propriedades e relações que descrevem cada conceito da taxonomia em um dicionário de conceitos.
5. **Definir relações *ad hoc* detalhadamente:** O objetivo desta tarefa é descrever detalhadamente todas as relações *ad hoc* binárias incluídas no dicionário de conceitos.
6. **Definir atributos de instância detalhadamente:** O ontologista descreve em detalhe todos os atributos de instância já inclusos no dicionário de conceitos através de uma tabela de atributos de instância.
7. **Definir atributos de classe detalhadamente:** O objetivo desta tarefa é descrever todos os atributos de classe já inclusos no dicionário através de uma tabela de atributos de classe.
8. **Definir constantes em detalhe:** Descreve com detalhes cada uma das constantes definidas no glossário de termos.
9. **Definir axiomas formais:** O ontologista identifica os axiomas formais na ontologia e os descreve precisamente.
10. **Definir regras:** Similarmente à tarefa anterior o ontologista identifica as regras e as descreve em uma tabela de regras.
11. **Definir instâncias:** O ontologista define instâncias relevantes que aparecem no dicionário de conceitos dentro de uma tabela de instâncias.

Estes passos serão aplicados futuramente na construção de uma ontologia para um sistema de recuperação de imagens.

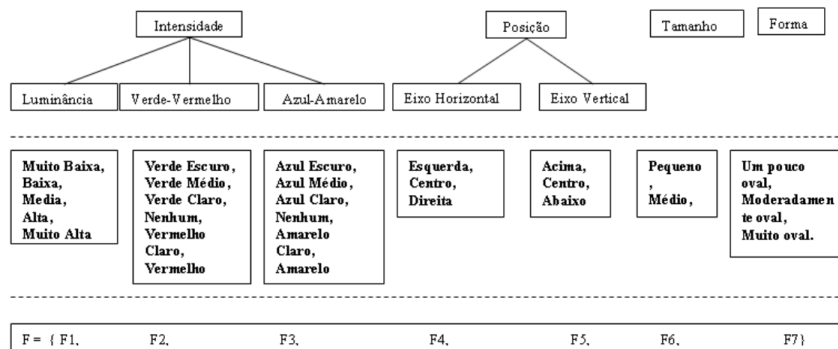
### 3.3 TRABALHOS RELACIONADOS ENVOLVENDO ONTOLOGIAS E CBIR

Alguns trabalhos na literatura abordam a combinação de ontologias e técnicas de CBIR para a diminuição do *semantic* e do *sensorial gap*. Em Mezaris, Kompatsiaris e Strintzis (2004) define-se uma metodologia envolvendo a chamada *ontologia de objeto*.

A ontologia de objeto define, ao invés de conceitos relativos a um domínio específico, um conjunto de atributos capazes de serem extraídos através de descritores de imagem.

A Figura 17 exemplifica uma ontologia de objeto. Na parte inferior, tem-se um vetor resultante da aplicação de descritores em uma imagem qualquer. Este vetor  $F$  é obtido a partir da tupla  $\varepsilon_D$  de um conjunto de descritores  $D$ , onde  $F1 =$  luminância,  $F2 =$  verde,  $F3 =$  vermelho,  $F4 =$  azul,  $F5 =$  Posição,  $F6 =$  Tamanho,  $F7 =$  Forma. Estes são os chamados “atributos de baixo nível” obtidos diretamente da aplicação dos descritores.

No topo da Figura 17 estão os conceitos da ontologia de objeto. Estes representam atributos obtidos através dos vetores de características dos descritores. Para cada conceito há um conjunto de valores, representados sob a forma de indivíduos e que se encontram na parte central da mesma figura.



**Figura 17: Exemplo de ontologia de objeto**

**Fonte: (FRANCISCANI, 2013)**

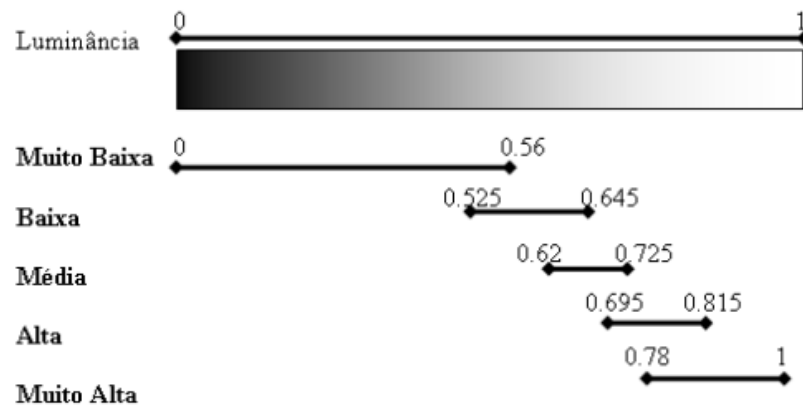
O mapeamento entre os atributos de baixo-nível para indivíduos da ontologia segue o processo apresentado na Figura 18, que ilustra o procedimento para o atributo de luminância. Regiões cujo valor de luminância possua mais de um valor classificam-se como mais de um indivíduo na ontologia de objeto (por exemplo, para uma luminância de 0,55 escolhem-se os atributos “Muito baixa” e “Baixa”).

### 3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo teve como objetivo apresentar os conceitos relativos às ontologias, seus métodos de construção, com ênfase no *Methontology*, utilizado neste trabalho. Foram apresentados alguns trabalhos envolvendo a utilização de ontologias e técnicas de CBIR aplicadas na recuperação de imagens.

A metodologia empregada neste trabalho é diferente desta, sendo descrita na Seção 5.4.





**Figura 18: Processo de mapeamento entre atributos de baixo nível para indivíduos da ontologia de objeto**

**Fonte: (FRANCISCANI, 2013)**

## 4 FERRAMENTA DE SUPORTE AOS TESTES EXPERIMENTAIS

A ferramenta desenvolvida tem como objetivo apoiar a execução de experimentos envolvendo CBIR e Ontologias, e possui os mesmos componentes típicos discutidos na Seção 2.5. As seções abaixo detalham cada um dos componentes da ferramenta de apoio aos experimentos, sendo eles módulo de processamento (seção 4.1), interface (seção 4.2) e base de imagens (4.3).

### 4.1 MÓDULO DE PROCESSAMENTO

O módulo de processamento da ferramenta desenvolvida baseia-se no conceito de experimento, definido como um conjunto de consultas (*queries*) de teste em uma base de treinamento. Desta forma, pode-se executar de maneira automática várias buscas para a análise de uma combinação de descritores. Portanto, pode-se dizer que o módulo de processamento é “orientado a experimentos”, apresentando as seguintes características:

1. permite gerar um experimento a partir de uma base de imagens e persistí-lo em XML;
2. permite utilizar mais de uma *query* por execução e fazer a análise dos resultados de precisão e revocação;
3. permite gerar uma saída personalizada, com os resultados do experimento, persistida em arquivo texto;

A geração de um experimento consiste em criar um arquivo de especificação, o qual contém o conjunto de entradas de treinamento e teste. Como isso consiste em um processo de classificação supervisionada, a base utilizada nos experimentos precisa estar previamente classificada, ou seja, cada amostra deve ter associado um rótulo que identifique a qual classe ela pertence.

A execução do experimento consiste em abrir o arquivo de especificações, executá-lo a partir das informações apresentadas neste último e gerar os arquivos de saída, contendo os resultados, representados por tabelas de precisão e revocação ou matrizes confusão, por exemplo.

Os vetores de características das imagens da base de testes são calculados previamente e persistidos em arquivos de texto plano. Este processo de cálculo é representado como um caso de uso próprio, apresentado na Seção 6.2.3. Ele consiste em abrir toda a base de imagens e para cada descritor de um conjunto definido, calculá-lo para cada entrada, salvando o vetor em uma pasta específica. O vetor de características relativo a uma imagem possui o mesmo nome da mesma e como extensão o nome do descritor. Para a imagem “segmim\_01781.jpg”, por exemplo, o valor do descritor DFC da mesma será persistido no arquivo “segmim\_01781.dfc” na pasta de vetores de características. Durante um processo de busca, a ferramenta identificará o arquivo de descritor associado a uma imagem através do padrão definido. Para agilizar os experimentos, permitiu-se, através de parâmetros, utilizar vetores já calculados também para as *queries*.

A personalização de saída do experimento é uma característica não funcional do sistema e permite, para um conjunto de dados do experimento, gerar saídas para facilitar a análise dos dados, como arquivos em texto plano, gráficos ou tabelas no padrão  $\text{\LaTeX}$ .

As seções abaixo descrevem melhor estas três características funcionais da ferramenta desenvolvida: criação de um experimento (seção 4.1.1), execução do mesmo (seção 4.1.2) e personalização de suas saídas (seção 4.1.3).

#### 4.1.1 CRIAÇÃO DE UM EXPERIMENTO

A criação de um experimento pode ser dividida em duas etapas: classificação da base e elaboração do arquivo de especificação do experimento. A classificação da base é um processo manual (ou seja, não é realizada automaticamente pela ferramenta), realizado separando-se as imagens da base em classes, as quais devem ser agrupadas em pastas do sistema operacional. Estas são utilizadas para gerar o arquivo de especificação.

Suponha uma base composta de 220 imagens, correspondentes a 12 classes de objetos. A classificação consiste em separar o diretório que as contém em 12 pastas, referentes às classes. A Figura 19 ilustra o processo.

Com a base devidamente separada, é possível executar a etapa de criação do arquivo de experimento, realizada automaticamente pela ferramenta.

Para cada diretório de classe de objetos, 20% das imagens são atribuídas como entradas de teste e os 80 % restantes como entradas de treinamento. Em seguida, o arquivo de especificação de experimento é definido sob a linguagem de marcação XML. A Figura 20 ilustra um exemplo.

Esse arquivo XML possui a seguinte estrutura:



**Figura 19: Separação das classes de flores em pastas**

- *experiment*: é a marcação que define o experimento;
- *class\_elements*: define uma classe de imagem. Todas as imagens de uma classe estarão abaixo desta marcação;
- *training\_set*: é a marcação responsável por definir as imagens que são de treinamento;
- *test\_set*: define as imagens de teste;
- *entry*: define uma imagem, podendo ser de treinamento ou teste.

O arquivo de especificação do experimento é utilizado como entrada para sua execução de fato. Ele permite visualizar com detalhes as entradas de treinamento e testes, podendo ser manualmente editado pelo usuário para atender às suas necessidades.

#### 4.1.2 EXECUÇÃO DO EXPERIMENTO

A execução do experimento é uma característica funcional da ferramenta e pode ser dividida em 3 etapas: carregamento do arquivo de experimentos, execução das *queries* e cálculo das informações de saída do experimento.

O carregamento do arquivo tem como função a configuração do experimento de acordo com a especificação contida no arquivo XML, ou seja, o processo realiza a leitura do arquivo buscando as imagens que serão de treinamento e as que serão de teste. Tais informações são localizadas através das marcações XML.

Após a leitura do arquivo, objetos são criados para representar as imagens de teste e treinamento do experimento, pois assim não se torna mais necessário a constante leitura em arquivo, mas apenas a consulta ao objeto.

Os objetos criados são então manipulados por algoritmos que automaticamente geram *queries* aleatórias, ou seja, o processo seleciona imagens do conjunto de teste e as usa para fazer a validação de descritores, busca CBIR ou busca por ontologia.

```

<?xml version="1.0" encoding="UTF-8"?>
<experiment id="0">
  <class_elements name="classe3">
    <training_set>
      <entry>
        flowerDatabase/classe3/segmim_00608.jpg
      </entry>
    </training_set>

    <test_set>

      <entry>

        flowerDatabase/classe3/segmim_00658.jpg
      </entry>
      <entry>

        flowerDatabase/classe3/segmim_00720.jpg
      </entry>
    </test_set>
  </experiment>

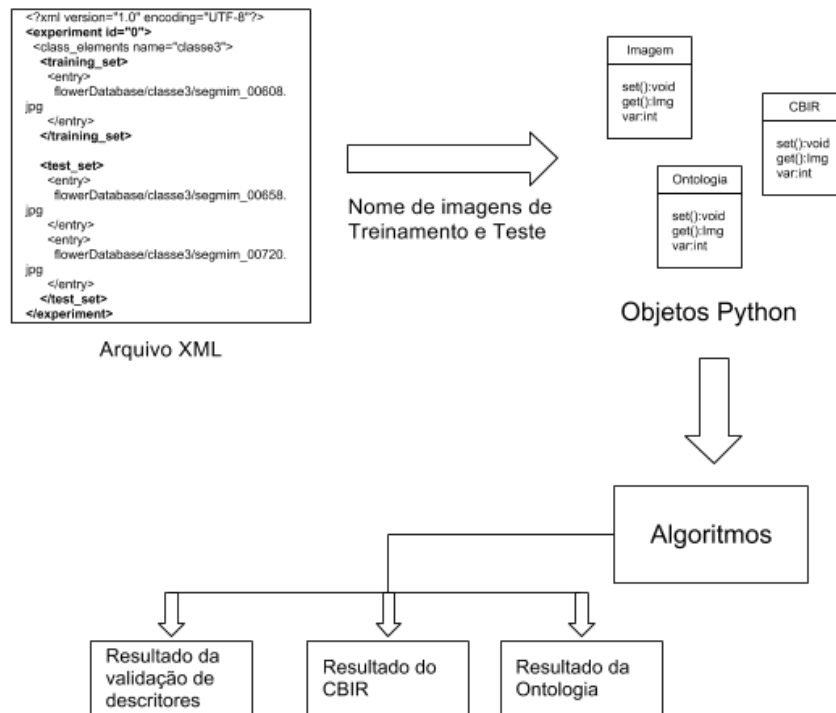
```

**Figura 20: Arquivo de marcação XML do experimento**

Cada *query* tem o seu descritor extraído, o qual é posteriormente comparado aos descritores que estão salvos no repositório de descritores. Tal repositório é obtido através da extração de descritores realizada através de uma opção na ferramenta, assim como discutido na Seção 4.2.

A saída do experimento está associada a qual busca foi executada. Desse modo, a Figura 21 ilustra três possíveis saídas do experimento: saída para validação de descritores, saída para busca CBIR e saída para ontologia.

As informações contidas na saída são as seguintes: arquivo contendo a matriz de confusão que é responsável por mostrar as classificações errôneas ocorridas no processo de validação de descritores, arquivo contendo a precisão atingida pelo processo de classificação KNN na validação de descritores, arquivo contendo a precisão e revocação para experimentos CBIR e de ontologia.



**Figura 21: Diagrama de execução de experimento**

#### 4.1.3 PERSONALIZAÇÃO DAS SAÍDAS DO EXPERIMENTO

A possibilidade de personalização das saídas é outra característica do sistema. Ela é realizada a partir da utilização do padrão DAO, descrito com detalhes na Seção 6.1.3.2.

O objetivo da personalização de saídas do experimento é reduzir o tempo de formatação e análise dos resultados. Por exemplo, para a realização da validação de descritores a classe empregada é a *ExperimentDAOTable*, que persiste os resultados de experimento na forma de tabelas de precisão e revocação e matriz confusão usando a sintaxe  $\LaTeX$ .

A Figura 22 ilustra a saída personalizada para a tabela de precisão e revocação de um experimento qualquer. O código gerado está pronto para ser anexado em um documento  $\LaTeX$  sem a necessidade de formatação. A outra saída personalizada é a matriz de confusão, conforme ilustrado na Figura 23.

#### 4.2 INTERFACE

São duas as interfaces principais: uma textual para criar e executar experimentos e outra gráfica para realização de consultas e recuperação de imagens por conteúdo.

```

\begin{table}[!h]
\centering
\begin{tabular}{|c|c|c|}
\hline Classe & Precisão & Revocação \\
\hline classe8 & 0.70125342734 & 0.789473684211 \\
\hline classe1 & 0.414634146341 & 0.733333333333 \\
\hline classe2 & 0.408299998291 & 0.825 \\
\hline classe3 & 0.803409825469 & 0.875 \\
\hline classe4 & 0.396415770609 & 0.571428571429 \\
\hline classe5 & 0.298062397373 & 0.625 \\
\\
\hline classe6 & 0.513888888889 & 0.5 \\
\hline classe7 & 0.319074224672 & 0.9375 \\
\hline classe12 & 0.49537037037 & 1.0 \\
\hline classe9 & 0.329511068642 & 0.692307692308 \\
\hline classe10 & 0.363416843408 & 0.942307692308 \\
\\
\hline classe11 & 0.814285714286 & 0.4625 \\
\hline
\end{tabular}

\caption{Tabela de precisão e revocação para o experimento}
\end{table}

```

**Figura 22: Arquivo de marcação XML do experimento**

#### 4.2.1 INTERFACE TEXTUAL: EXPERIMENTOS

Por meio da interface, ilustrada na Figura 24, é possível realizar diversas tarefas, incluindo aquelas relacionadas aos experimentos e à extração de vetores de característica.

A sua principal vantagem é a execução rápida de tarefas relativas ao experimento, cujos resultados são armazenados em pastas de arquivos contendo o resultado das *queries* e suas estatísticas.

A opção 1 executa uma consulta individual, ou seja, para uma única imagem. Esta imagem deve estar no diretório */tmp* com o nome de *query.jpg*. A opção 2 realiza a criação do experimento, como discutido na Seção 4.1.1.

A extração dos vetores de característica das imagens da base é realizada através da opção 3. A Figura 25 ilustra a saída da ferramenta ao escolher a opção.

A opção 4, abre um experimento e exhibe na tela as entradas de treinamento e teste, para breve conferência. A opção 5 executa um experimento. Como discutido na Seção 4.1.2, a

```

\begin{table}[!h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline Classe & classe3 & classe4 & classe1 & classe2 & classe5 & \\
\hline classe3 & 4 & 0 & 0 & 0 & 0 & \\
\hline classe4 & 0 & 4 & 0 & 0 & 0 & \\
\hline classe1 & 0 & 0 & 3 & 0 & 0 & \\
\hline classe2 & 0 & 0 & 0 & 5 & 0 & \\
\hline classe5 & 0 & 0 & 0 & 1 & 3 & \\
\hline
\end{tabular}
\caption{Matriz de confusão do experimento}
\end{table}

```

**Figura 23: Arquivo de marcação XML do experimento**

execução do experimento abre o arquivo de especificação do experimento, executando uma *query* para cada entrada de teste e armazenando as respostas em disco. A Figura 26 ilustra a saída da ferramenta para o momento de execução do experimento.

A opção 6 permite a execução de um experimento envolvendo ontologias, que consiste em carregar o arquivo de experimento e executá-lo nos moldes de um experimento CBIR, como detalhado na seção 5.4.

No caso da validação de descritores, há a necessidade de segmentar as imagens da base através da opção 7, para que facilite a recuperação do descritor. A segmentação de imagem retira regiões que não são de interesse para que estas não interfiram na extração dos descritores.

#### 4.2.2 CBIR

No processo de CBIR implementado na ferramenta desenvolvida, é possível realizar uma *query* individual através de uma interface *web*, que pode ser vista na Figura 27. É solicitado ao usuário a imagem ele deseja utilizar para consulta e o sistema é chamado após a confirmação.

As respostas da busca serão salvas da mesma forma da opção 1 da interface textual (*query* individual), nas pastas do sistema operacional. Os detalhes dos parâmetros de execução e as saídas do experimento baseado em descritores serão descritos no Capítulo 5.



```

/home/solano/tcc2/Python-Descriptors/RunOffline.py
Finalizado!
Insira:
 1 para query individual |
 2 para criar experimento
 3 para calcular descritores
 4 para abrir experimento
 5 para executar experimento
 6 para executar experimento ontologias
 7 para segmentar a base
Valor:

```

**Figura 24: Interface para validação de descritores**

```

 3 para calcular descritores
 4 para abrir experimento
 5 para executar experimento
 6 para executar experimento ontologias
 7 para segmentar a base
Valor:3
Calculating dcd ...
Calculating flowerDatabase/classe11/segmim_04922.jpg

Calculating flowerDatabase/classe11/segmim_04928.jpg

Calculating flowerDatabase/classe11/segmim_04918.jpg

```

**Figura 25: Saída do cálculo de descritores**

### 4.2.3 ONTOLOGIA

No que se refere ao uso de ontologias, as *queries* podem ser realizadas através da interface textual. Um exemplo de execução de experimento é apresentado na Figura 28. Tal execução utiliza a opção 6 da interface textual.

Os detalhes dos parâmetros de execução e as saídas do experimento baseado ontologias serão descritos no Capítulo 5.

### 4.3 BASE DE IMAGENS

Uma das principais preocupações deste trabalho diz respeito à escolha da base de imagens. Devido à necessidade de realização de um estudo comparativo entre técnicas envolvendo CBIR e ontologias, foi necessário que aquela atendesse aos seguintes requisitos:

---

```

4 para abrir experimento
5 para executar experimento
6 para executar experimento ontologias
7 para segmentar a base
Valor:5
Running dcd ...
Calculating dcd for the query image flowerDatabase/classe3/segmim_00658.jpg
dcd calculated!

Running sift ...
Calculating sift for the query image flowerDatabase/classe3/segmim_00658.jpg

```

**Figura 26: Exemplo de execução de experimento**

1. Executar experimentos usando técnicas de CBIR
2. Apresentar um conjunto de objetos capazes de serem conceitualizados por uma ontologia
3. Ser segmentada, visando melhorar o desempenho dos descritores de cor e forma.
4. Apresentar um número considerável de objetos, para facilitar a criação de bases controladas (BCs) que permitam analisar melhor a resposta de cada descritor. (As bases controladas são apresentadas no início do capítulo 5 ).

A base que atendeu a todos estes requisitos é a “102 *Flower Category dataset*” (NILS-BACK; ZISSERMAN, 2012), composta de 102 espécies de flores.

Esta base foi objetivo de estudo de dois trabalhos envolvendo técnicas de CBIR. Em Nilsback e Zisserman (2010) discute-se um método automático de segmentação das imagens da base envolvendo dois métodos: Um baseado em cor de fundo e região de interesse e outro baseado na estrutura das pétalas.

Em Nilsback e Zisserman (2008) é realizado um trabalho de classificação automática de flores envolvendo técnicas de CBIR. Ele assemelha-se ao experimento de CBIR realizado neste trabalho, porém utiliza a base completa (composta por 103 categorias de imagens) e ferramentas mais sofisticadas de classificação e ordenação dos resultados. Ele emprega quatro descritores : Cor dominante, SIFT da imagem segmentada e sem segmentação e o *HOG* (DALAL; TRIGGS, 2005), utilizando o SVM como classificador. Segundo os autores, a taxa de precisão dos experimentos obtida a partir dos experimentos realizados é de 72,8 %.

Como este trabalho utiliza um classificador mais simples (KNN) e a construção de uma ontologia, optou-se por reduzir o número de classes da base de imagens em quatro bases controladas. Essas bases tem como objetivo avaliar o desempenho de descritores, por questões discutidas no início do Capítulo 5.



**Figura 27: Tela inicial da ferramenta**

#### 4.4 VISÃO GERAL DOS COMPONENTES DO SOFTWARE

A ferramenta foi desenvolvida utilizando a tecnologia Python e segue o padrão de desenvolvimento intitulado *Model-View-Controller* (MVC) (REENSKAUG, 1979). Esse padrão separa o software em camadas que possuem cada uma um diferente papel.

Para a camada *Model*, há a agregação de acesso à dados. A camada *View* é responsável pela interface de comunicação do usuário com o software, podendo ela ser uma página da internet ou aplicativo para computador, por exemplo. Por fim a *Controller* é responsável pela camada de manipulação e lógica dos dados, que realiza as operações sobre dados e os exibem ao usuário (REENSKAUG, 1979).

A estrutura do software modelada no MVC pode ser vista na Figura 29, que também ilustra as classes principais que compõem o fluxo de execução de um experimento CBIR e por

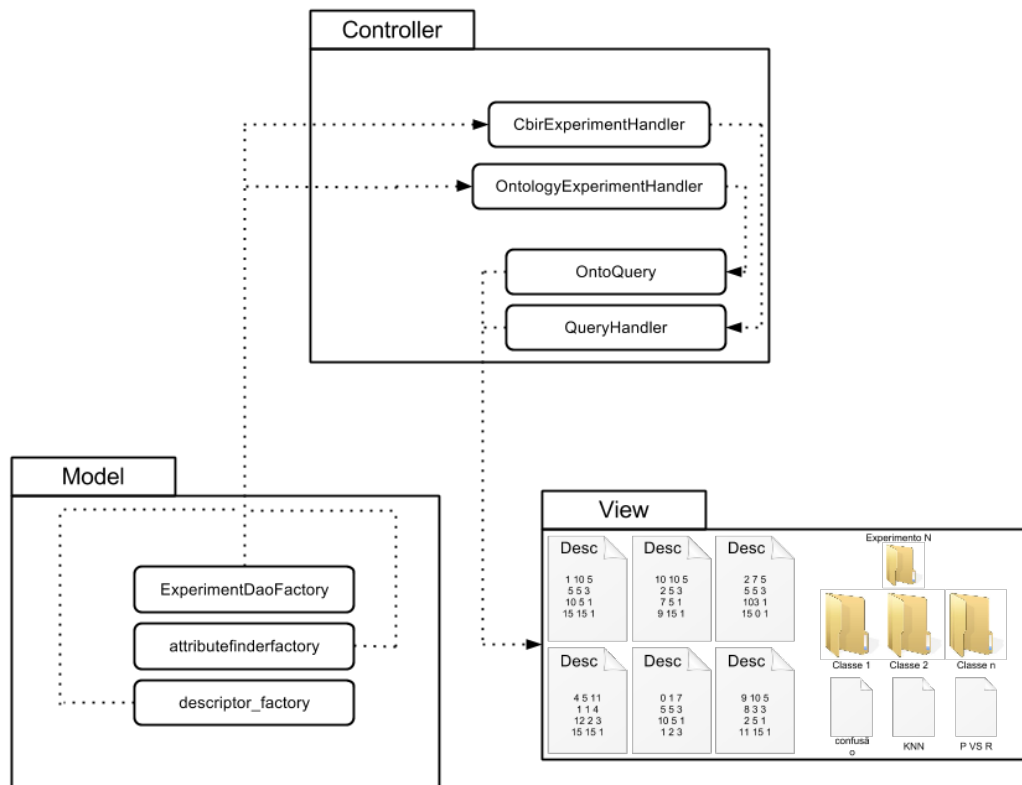
```

Finalizado!
Insira:
1 para query individual
2 para criar experimento
3 para calcular descritores
4 para abrir experimento
5 para executar experimento
6 para executar experimento ontologias
7 para segmentar a base
Valor:6
Passo 1 : Query por tamanho real (Insira I para valor igual, S para Semelhante e N para pular)
Opção:I
Passo 2 : Query por cor (Insira I para valor igual, S para Semelhante e N para pular)
Opção:S
Passo 3 : Query por localização (Insira I para valor igual, S para Semelhante e N para pular)
Opção:N
Passo 4 : União ou intersecção (Insira U para união ou I para intersecção)
Opção:U
Finalizado

```

**Figura 28: Exemplo de execução de experimento de ontologias**

ontologia.



**Figura 29: Software na perspectiva da metodologia Model View Controller**

Quando são utilizados os descritores de cor e forma para recuperação de imagens, o módulo *Model* age através da classe *descriptor\_factory*, a qual é responsável por recuperar os descritores que serão usados. Quando a ontologia é usada, o módulo *Model* age através da classe *attributefinderfactory* responsável pela recuperação de termos da ontologia que representam as características da imagem. A classe *ExperimentDaoFactory* faz a manipulação dos dados, ou seja, das imagens da base que serão comparadas uma a uma com a imagem de consulta para verificar a similaridade.

Como mencionado, a camada *Controller* é responsável por manipular as classes do *Model* e do *View*. Para a recuperação baseada em descritores, a classe *CbirExperimentHandler* é usada, a qual depende da classe *descriptor\_factory* e *ExperimentDaoFactory*. Após a manipulação dessas duas classes, o experimento é executado através da chamada da classe *QueryHandler*. Para o caso da ontologia, a classe *OntologyExperimentHandler* é utilizada, a qual depende da classe *attributefinderfactory* e da classe *ExperimentDaoFactory*. Após a manipulação dessas duas classes, o experimento é executado através da chamada da classe *OntoQuery*.

Os resultados ficam abaixo de uma única pasta, tanto quando são utilizados descritores quando ontologias. A saída é representada pela camada *View*, onde os dados podem ser visualizados pelo usuário da aplicação. Essa camada é responsável por mostrar os descritores gerados para cada imagem e também a árvore de diretórios - que para cada imagem de consulta, representada por uma pasta, exibe as imagens que foram classificadas como corretas e incorretas, em duas sub-pastas distintas. Os arquivos gerados são responsáveis por exibir os resultados.

#### 4.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

A ferramenta de apoio aos experimentos foi construída para facilitar a realização do estudo comparativo, o que é evidenciado através de seus três características principais. A definição de experimento como um conjunto de *queries* de teste em uma base de treinamento é fundamental para a concepção do *software* e seus requisitos.

A utilização de um arquivo de especificação de experimento gerado de forma automática permite à ferramenta trabalhar de forma rápida com mais de uma base de imagens, utilidade aproveitada na etapa de validação de descritores, quando se utilizam quatro bases diferentes (capítulo 5).

A execução de múltiplas *queries* em uma única vez facilita as etapas do estudo comparativo, do capítulo 5, quando são realizados vários experimentos nas etapas de validação, CBIR e Ontologias.

## 5 DESENVOLVIMENTO

O desenvolvimento deste trabalho foi realizado em três etapas, descritas ao longo deste capítulo: validação dos descritores (Seção 5.2), recuperação de imagens com base em descritores e proposta de apoio usando ontologias (Seções 5.3 - 5.5).

A Seção 5.1 descreve as bases de imagens consideradas nos testes e a Seção 5.6 discute as considerações deste estudo teórico.

### 5.1 BASE DE IMAGENS

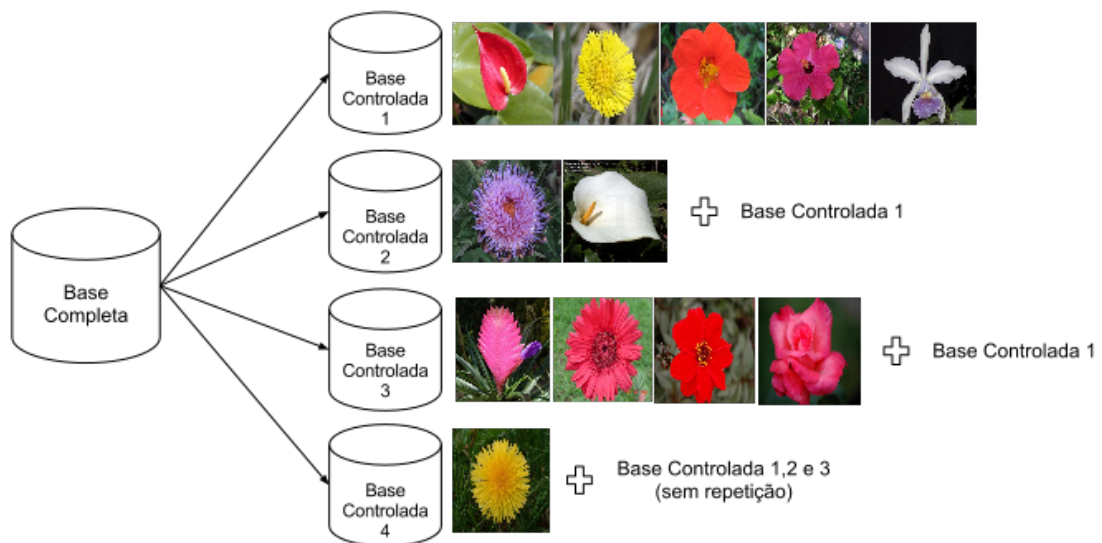
A base de imagens utilizada no estudo foi dividida em quatro bases controladas, cada uma com características específicas. O objetivo foi construir cenários de teste com diferentes níveis de dificuldade que permitissem analisar a discriminação feita pelos descritores, sendo eles SIFT e DFC para forma e DCD e GCH para cor.

Cada base controlada permite avaliar o desempenho dos descritores e identificar cenários onde os descritores tem pior ou melhor desempenho. A Figura 30 ilustra a distribuição das flores para cada base controlada.

A Tabela 3 especifica a quantidade de imagens existentes em cada classe.

#### 5.1.1 BASE CONTROLADA 1 (BC 1)

Essa base contém cinco classes de flores, ilustradas na Figura 31, as quais possuem características diferentes com relação à forma e cor (com base em uma análise perceptual por parte dos autores deste trabalho). Os experimentos com as imagens desta base visam avaliar a eficiência dos descritores utilizados, dado que diferenças significativas entre as classes tipicamente implicam em uma maior taxa de acerto, mesmo para descritores simples.



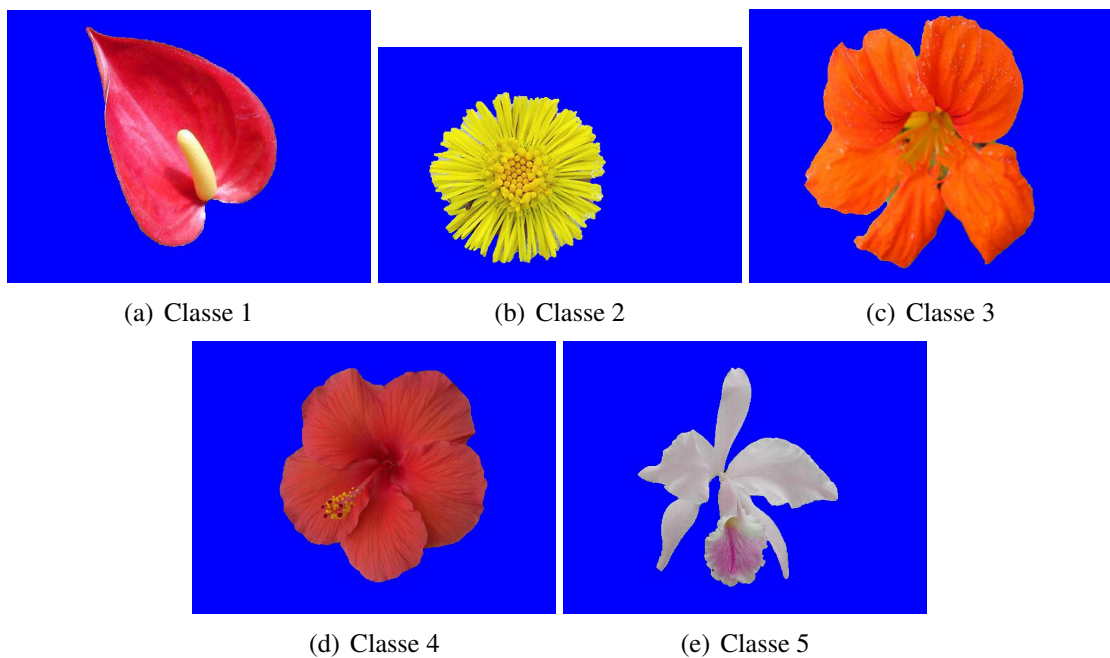
**Figura 30: Bases controladas usadas nos experimentos**

Classe	Total de imagens
classe1	18
classe2	21
classe3	18
classe4	18
classe5	16
classe6	16
classe7	16
classe8	24
classe9	17
classe10	17
classe11	21
classe12	17

**Tabela 3: Total de imagens por classe**

### 5.1.2 BASE CONTROLADA 2 (BC 2)

A BC 2 é composta pelas imagens que compõem a BC 1 juntamente com as classes 10 e 11, que possuem características de forma semelhantes às das classes 2 e 1, respectivamente. Portanto, o objetivo dessa base é avaliar a eficiência dos descritores para recuperação em uma base contendo flores com formas semelhantes. A Figura 32 ilustra amostras de cada uma das classes presentes na BC 2.



**Figura 31: BC 1 - Base com flores possuindo características distintas de forma e cor.**

### 5.1.3 BASE CONTROLADA 3 (BC 3)

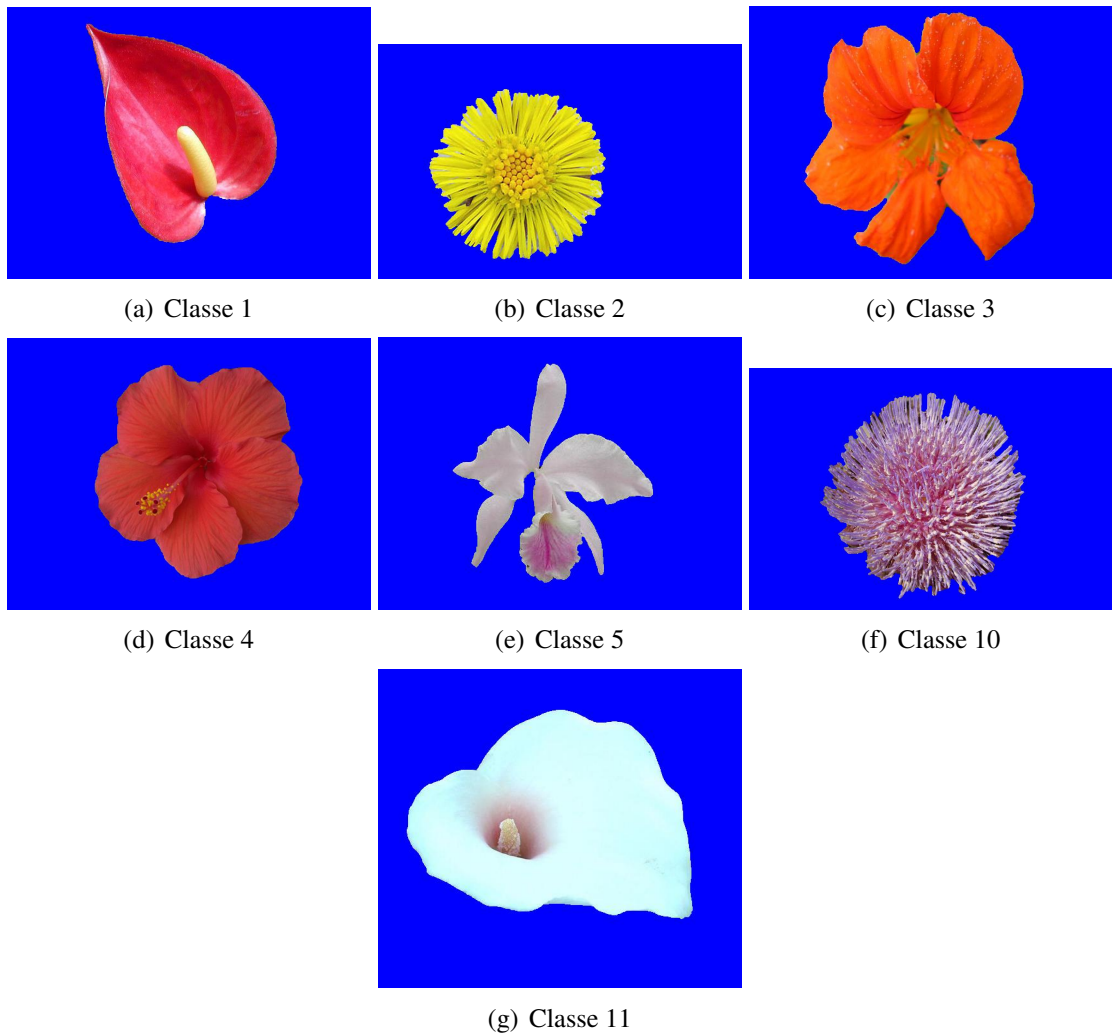
Visando avaliar os descritores de cor, a BC 3 é construída adicionando as classes 6, 7, 8 e 9 à BC 1, as quais possuem, respectivamente, cores semelhantes às das classes 4, 1, 3 e 4 (novamente com base em uma análise perceptual). Buscou-se que evitar que as características de forma entre as classes fossem muito semelhantes.

Sendo assim, a realização de experimentos com amostras dessa base visa avaliar os dois descritores de cor quando utilizados em um processo de classificação em uma base contendo classes com formas diferentes e cores semelhantes, ilustradas na Figura 33.

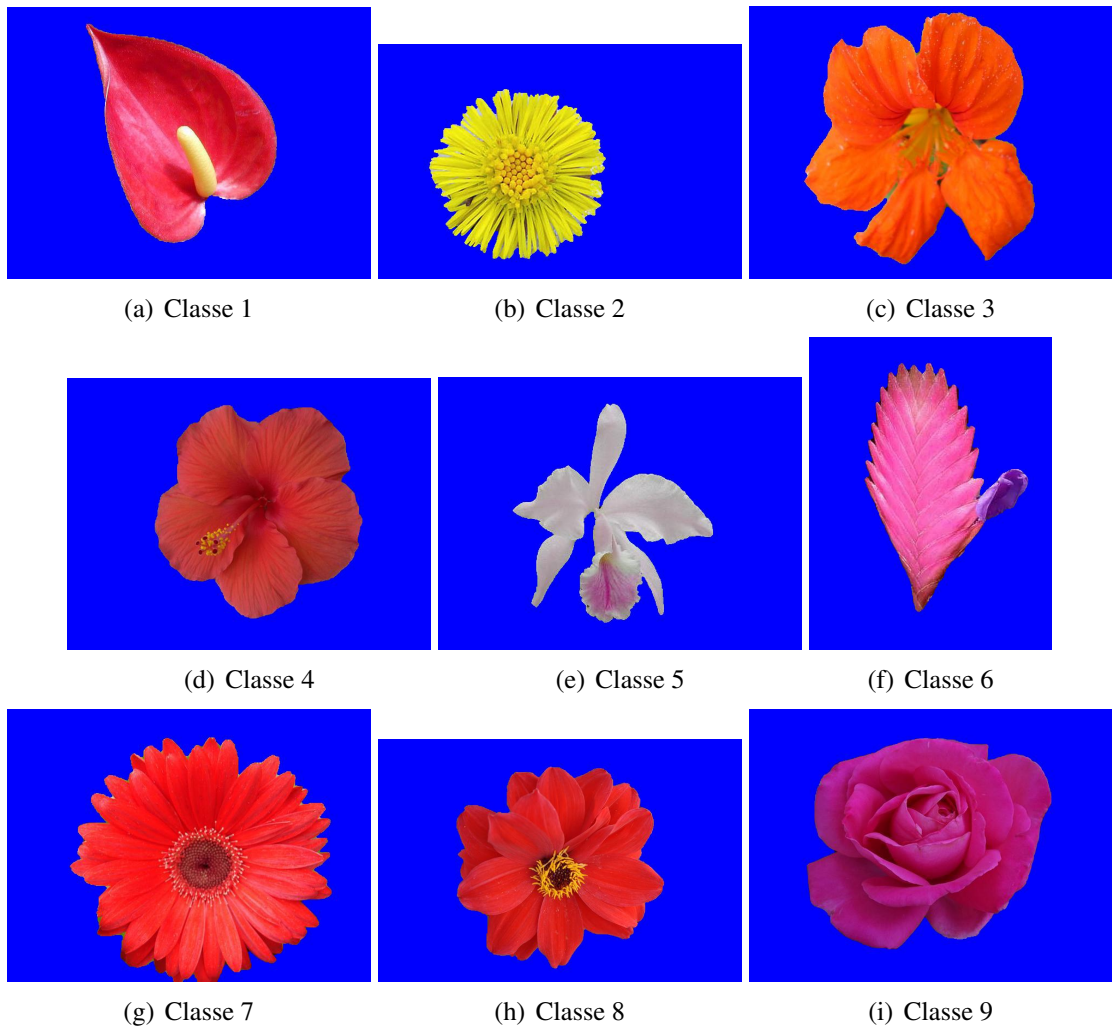
### 5.1.4 BASE CONTROLADA 4 (BC 4)

A BC 4 agrupa todas as classes de flores utilizadas na etapa de validação de descritores. O objetivo dessa base é criar um cenário mais similar a um real, que tipicamente envolve diferentes tipos de imagens não facilmente discrimináveis. A Figura 34 ilustra uma amostra para cada classe dessa base.

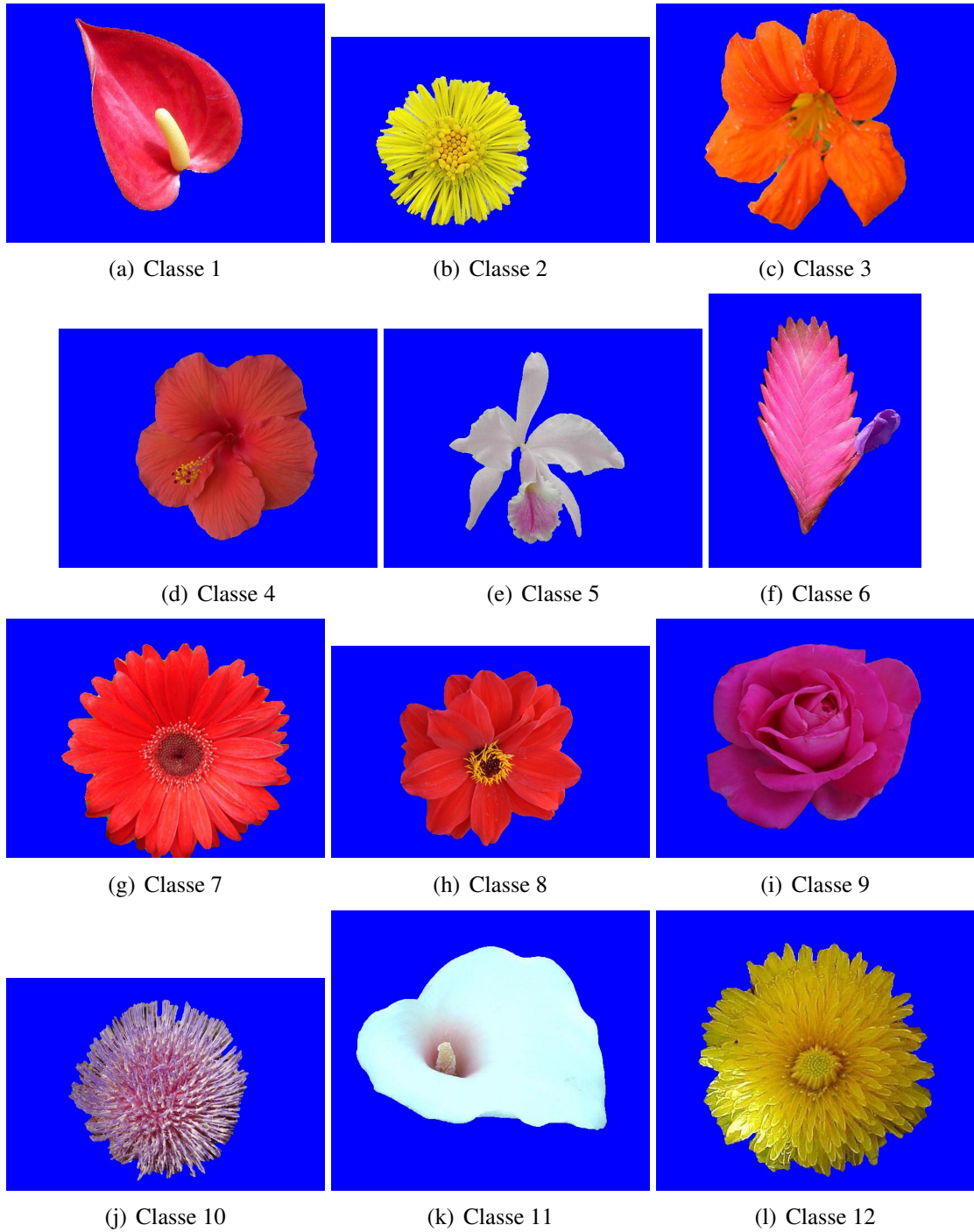




**Figura 32: BC 2 - Base com flores possuindo características distintas de cor e algumas semelhantes de forma.**



**Figura 33: BC 3 - Base com flores possuindo características semelhantes de cor e algumas diferentes de forma.**



**Figura 34: BC 4 - Base com flores possuindo características semelhantes de cor e forma.**

## 5.2 VALIDAÇÃO DOS DESCRITORES

Nesta seção, são analisados os desempenhos individual e das combinações dos quatro descritores utilizados (GCH, DCD, DFC e SIFT) em um processo de classificação supervisionada usando o algoritmo KNN. Entende-se por combinação a utilização de dois ou mais descritores em um mesmo processo de classificação. Esta etapa visa, além de validar a implementação, avaliar quais os descritores mais robustos, para então utilizá-los na recuperação de imagens propriamente dita.

A Figura 35 ilustra, resumidamente, o processo de extração de descritores de características para cada base controlada de testes. Como comentado anteriormente, no processo de classificação supervisionada proposto, as imagens de tais bases são divididas em dois subconjuntos, sendo 20% para teste e 80% para treinamento. A avaliação será realizada através da análise da medida de precisão e da matriz de confusão resultantes de cada experimento.



**Figura 35: Teste de descritores na base controlada n**

Para cada base de testes, são realizados oito experimentos, os quais representam a utilização de um vetor de características diferente no processo de classificação. Tais vetores são compostos pelo(s) descritor(es):

1. GCH
2. DCD
3. SIFT
4. DFC
5. GCH e DFC
6. GCH e SIFT
7. DCD e DFC
8. DCD e SIFT

Neste trabalho, o classificador utilizado é o KNN considerando-se  $K=7$  (através de testes empíricos, concluiu-se que este é o que apresenta melhor desempenho na classificação). Para que a combinação fosse possível, as saídas de cada um dos descritores foi normalizada para um valor entre 0 e 1. Não há alteração do resultado obtido quando são utilizados os valores originais, pois foi obtido o valor máximo e mínimo da distância para cada descritor, tornando possível o uso da normalização. Espera-se com esta etapa avaliar os acertos e erros de classificação, para determinar se as implementações de cada descritor atendem aos resultados esperados pela análise teórica.

### 5.2.1 AVALIAÇÃO DOS DESCRITORES PARA CLASSIFICAÇÃO DA BC 1

Como detalhado anteriormente, os experimentos com a BC 1 visam avaliar o desempenho de cada combinação de descritores em um processo de classificação supervisionada em uma base cujas imagens apresentam tanto características de cor quanto de forma distintas.

#### EXPERIMENTO 1 - GCH

A precisão para cada classe é mostrada na Tabela 4 e a matriz de confusão na Tabela 5.

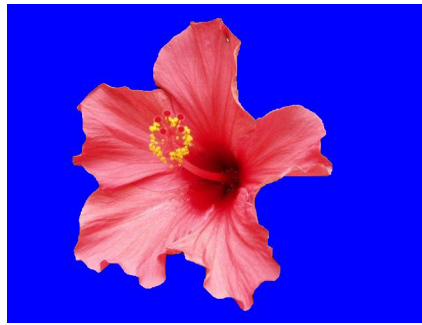
Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	75
classe5	100

**Tabela 4:** Precisão considerando o descritor GCH na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe3	4	0	0	0	0
classe4	0	3	1	0	0
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe5	0	0	0	0	4

**Tabela 5:** Matriz de confusão para o descritor GCH na BC 1

Observe (Tabela 4) que a taxa de acerto é de 100% para todos os casos, exceto a classe 4, que atingiu 75%. Com base na análise da matriz de confusão, mostrada na Tabela 5, é possível identificar que houve um erro de classificação que atribui a classe 1 à uma amostra pertencente à classe 4. Como tal amostra, ilustrada na Figura 36, possui cor próxima àquela presente na



**Figura 36: Amostra classificada erroneamente no experimento 1**

classe 1, pode-se concluir que o GCH apresenta erros quando imagens de cores próximas estão em classes diferentes, o que é coerente com a forma que o descritor é implementado.

## EXPERIMENTO 2 - DCD

Ao considerar o DCD, obteve-se uma taxa de precisão de 100% para todas as classes exceto a 5 (Tabela 6). Isso ocorre devido à existência de tons de cores diferentes do branco, a princípio dominante nesta classe, o que levou a um erro de classificação com atribuição de uma amostra da classe 5 à classe 2, conforme pode-se observar na Tabela 7.

Pode-se concluir que o descritor DCD implementado sofre com interferências provocadas por imagens com vários tons de cor, sendo mais adequado aos casos que a imagem possui uma tonalidade dominante ou poucos tons de cores distintas. Como nem sempre é possível ter controle sobre isso, este fato mostra a necessidade de fazer o processo de recuperação baseado em mais de um tipo de característica.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	75

**Tabela 6: Precisão considerando o descritor DCD na BC 1**

Classe	classe3	classe4	classe1	classe2	classe5
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe3	4	0	0	0	0
classe4	0	4	0	0	0
classe5	0	0	0	1	3

**Tabela 7: Matriz de confusão para o descritor DCD na BC 1**

### EXPERIMENTO 3 - SIFT

O descritor SIFT teve 100% de precisão para todas as classes (Tabela 8), ou seja, não houve nenhum erro de classificação, como mostra a Tabela 9.

Classe	Precisão (%)
classe3	100
classe4	100
classe1	100
classe2	100
classe5	100

**Tabela 8:** Precisão considerando o descritor SIFT na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe3	4	0	0	0	0
classe4	0	4	0	0	0
classe5	0	0	0	0	4

**Tabela 9:** Matriz de confusão do experimento utilizando SIFT na BC 1

### EXPERIMENTO 4 - DFC

O descritor DFC mostrou melhor desempenho para a classe 1, conforme ilustra a Tabela 10, caracterizada por possuir forma pontiaguda. Como é a única a apresentar este padrão, possibilita uma boa discriminação em relação às outras classes da base.

A classe 2 teve apenas um erro de classificação, ilustrado na Tabela 11, resultando em uma precisão de 80%. Este erro deve-se à classificação de uma amostra como pertencente à classe 2, o que é justificado pela forma arredondada das duas.

As classes 3 e 4 apresentaram um número maior de erros devido à forma arredondada gerada pelo feixo convexo ao extrair o contorno da imagem. Isso ocorre porque este descritor também gera formas arredondadas para amostras das classes 1, 2 e 3.

A forma de pétalas espaçadas da classe 5 influenciou o algoritmo a confundir a classificação desta com a forma arredondada e pétalas espaçadas da classe 4 gerando uma precisão de 75% para a classe 5.

Pode-se concluir que o DFC é sensível à forma gerada pelo feixo convexo, pois uma vez que a forma se assimila ao padrão arredondado, há chances de erro de classificação.

Classe	Precisão (%)
classe3	50
classe4	25
classe1	100
classe2	80
classe5	75

**Tabela 10:** Precisão considerando o descritor DFC na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe3	2	1	1	0	0
classe4	1	1	1	1	0
classe1	0	0	3	0	0
classe2	0	1	0	4	0
classe5	0	1	0	0	3

**Tabela 11:** Matriz de confusão do experimento utilizando DFC na BC 1

#### EXPERIMENTO 5 - GCH E DFC

Analisando-se as Tabelas 12 e 13, pode-se concluir que a composição obteve as mesmas taxas de precisão do GCH isolado (Tabela 4). Se por um lado isso é ruim, por outro mostra que a combinação não afeta o poder discriminativo do GCH.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	75
classe5	100

**Tabela 12:** Precisão considerando os descritores GCH e DFC na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe3	4	0	0	0	0
classe4	0	3	1	0	0
classe5	0	0	0	0	4

**Tabela 13:** Matriz de confusão considerando os descritores GCH e DFC na BC 1

#### EXPERIMENTO 6 - GCH E SIFT

A partir das informações das Tabelas 14 e 15, o desempenho é equivalente àquele obtido usando-se apenas o SIFT (Tabela 8).



Classe	Precisão (%)
classe3	100
classe4	100
classe1	100
classe2	100
classe5	100

**Tabela 14:** Precisão considerando os descritores GCH e SIFT na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe3	4	0	0	0	0
classe4	0	4	0	0	0
classe5	0	0	0	0	4

**Tabela 15:** Matriz de confusão do experimento para os descritores GCH e SIFT na BC 1

Isso mostra que, embora o GCH não interfira no poder discriminativo do SIFT, não há vantagem nenhuma em utilizá-lo neste caso.

#### EXPERIMENTO 7 - DCD E DFC

Como pode-se observar nas Tabelas 16 e 17, a composição DCD+DFC seguiu o padrão do DCD isolado.

Classe	Precisão (%)
classe3	100
classe4	100
classe1	100
classe2	100
classe5	75

**Tabela 16:** Precisão considerando os descritores DCD e DFC na BC 1

Classe	classe3	classe4	classe1	classe2	classe5
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe3	4	0	0	0	0
classe4	0	4	0	0	0
classe5	0	0	0	1	3

**Tabela 17:** Matriz de confusão considerando os descritores DCD e DFC na BC 1

#### EXPERIMENTO 8 - DCD E SIFT

Assim como foi o caso para o Experimento 6, o DCD não interferiu na discriminação intra- ou inter-classe do SIFT (veja Tabelas 18 e 19).

Classe	Precisão (%)
classe3	100
classe4	100
classe1	100
classe2	100
classe5	100

**Tabela 18:** Precisão considerando os descritores DCD e SIFT na BC 1

#### 5.2.2 AVALIAÇÃO DOS DESCRITORES PARA CLASSIFICAÇÃO DA BC 2

Como detalhado anteriormente, os experimentos com a BC 2 visam avaliar o desempenho de cada combinação de descritores em um processo de classificação supervisionada em uma base cujas imagens apresentam características semelhantes de cor e distintas de forma.

Classe	classe3	classe4	classe1	classe2	classe5
classe3	4	0	0	0	0
classe4	0	4	0	0	0
classe1	0	0	3	0	0
classe2	0	0	0	5	0
classe5	0	0	0	0	4

**Tabela 19:** Matriz de confusão do experimento para os descritores DCD e SIFT na BC 1

#### EXPERIMENTO 1 - GCH

Neste experimento, as classes 4 e 10 apresentaram desempenho abaixo de 100% (Tabela 4) porque possuem classes correspondentes na base com tonalidades próximas. Portanto, os erros devem-se à proximidade de cores entre as classes, como pode ser vista na Tabela 5.

Como o GCH baseia-se apenas na cor, estes erros são esperados. Isso mostra a necessidade de utilizar a composição de descritores que consideram diferentes características.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	50
classe5	50
classe10	75
classe11	100

**Tabela 20:** Precisão considerando o descritor GCH na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	2	0	1	0	1	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	1	0	0	3	0
classe5	0	0	2	0	0	0	2

**Tabela 21:** Matriz de confusão para o descritor GCH na BC 2

Observe que o desempenho para a classe 5 caiu de 100% (na BC 1 - Tabela 4) para 50 %. Isso deve-se à inclusão da classe 11, que possui amostras em que a cor branca influencia na classificação de amostras da classe 5.

## EXPERIMENTO 2 - DCD

No caso do DCD, a adição de novas classes de flores só causou problemas na classe 5 (Tabela 22). Assim como no experimento anterior, tal erro se deve à classificação de uma amostra da classe 5 como pertencente à classe 11, que em algumas amostras tem a mesma tonalidade branca. A matriz de confusão é ilustrada na Tabela 23.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	75
classe10	100
classe11	100

**Tabela 22:** Precisão considerando o descritor DCD na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	4	0	0	0	0	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	0	0	0	4	0
classe5	0	0	1	0	0	0	4

**Tabela 23:** Matriz de confusão para o descritor DCD na BC 2

Neste contexto, pode-se concluir que o DCD é mais robusto que o GCH, pois conduziu a menos erros de classificação na inclusão de classes contendo algumas amostras com tons de cores semelhantes àqueles já existentes.

## EXPERIMENTO 3 - SIFT

Nesse experimento, diversos erros foram causados pela inserção da classe 11, conforme ilustram as Tabelas 24 e 25. Isso se deve ao fato da classe 11 possuir flores de formas variadas o que comprometeu a eficiência dos pontos de interesse encontrados pelo SIFT.

Os erros associados às classes 2 e 10 devem-se à similaridade entre elas (ressalta-se que a classe 10 foi inserida justamente com o objetivo de verificar tal impacto).

Classe	Precisão (%)
classe1	100
classe2	40
classe3	75
classe4	75
classe5	100
classe10	50
classe11	100

**Tabela 24:** Precisão considerando o descritor SIFT na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	3	0	1	0	0	0	0
classe4	0	3	1	0	0	0	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	3	0	2	0	0
classe10	0	0	2	0	0	2	0
classe5	0	0	0	0	0	0	4

**Tabela 25:** Matriz de confusão para o descritor SIFT na BC 2

#### EXPERIMENTO 4 - DFC

Como esperado, as classes 1 e 11, que possuem formas semelhantes, apresentaram baixas taxas de reconhecimento. Amostras da classe 11 foram classificadas como pertencentes à classe 5 devido à principal limitação da representação do DFC. Como o feixo convexo é representado como um “contorno” de pontos, há flores cuja forma acaba representada por uma aproximação da forma original, onde podem existir características de curvatura mais específicas.

A classe 5, por sua vez, aumentou a taxa de reconhecimento em 25% em comparação aos testes usando a BC 1. Isso se deve ao fato que a inclusão de mais classes possibilitou uma melhor discriminação entre essa classe e as demais ao considerar o DFC.

Classe	Precisão (%)
classe1	66
classe2	20
classe3	50
classe4	25
classe5	100
classe10	25
classe11	40

**Tabela 26:** Precisão considerando o descritor DFC na BC 2

Neste contexto, as classes 2, 3 e 10, que apresentam a forma arredondada, serão prejudica-

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	2	0	1	1	0	0	0
classe4	2	1	0	0	0	0	1
classe11	0	0	2	1	0	0	2
classe1	0	0	0	2	0	1	0
classe2	1	0	0	1	1	2	0
classe10	1	0	0	1	1	1	0
classe5	0	0	0	0	0	0	4

**Tabela 27:** Matriz de confusão do experimento utilizando DFC na BC 2

das no desempenho da classificação exata. Porém se o objetivo de alguma aplicação específica fosse recuperar imagens com formatos próximos, mesmo que pertencentes à outras classes, o desempenho do descritor seria melhor.

#### EXPERIMENTO 5 - GCH E DFC

O objetivo desse experimento foi avaliar a composição dos descritores GCH e DFC como forma de verificar o desempenho na classificação nessa base. As Tabelas 28 e 29 ilustram que esta composição elevou o desempenho dos descritores isolados.

Este resultado mostra que é possível explorar as informações complementares de diferentes características de cor e forma. Em comparação ao GCH, que obteve melhor desempenho que DFC quando considerado isoladamente na BC 2, observa-se que houve uma melhora na precisão para a classe 5.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	50
classe5	100
classe10	75
classe11	100

**Tabela 28:** Precisão considerando os descritores GCH e DFC na BC 2

#### EXPERIMENTO 6 - GCH E SIFT

Com base nas Tabelas 30 e 31, pode-se concluir que a composição elevou o desempenho desses descritores isolados. Assim como foi o caso do experimento anterior, este resultado mostra que é possível explorar as informações complementares de diferentes características para elevar as taxas de precisão.

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	3	0	0	0	1	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	1	0	0	3	0
classe5	0	0	0	0	0	0	4

**Tabela 29:** Matriz de confusão considerando os descritores GCH e DFC na BC 2

Em comparação ao GCH, que obteve melhor desempenho que SIFT quando considerado isoladamente na BC 2, observa-se um aumento nas taxas de acerto da classificação das classes 4 e 5.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	75
classe5	100
classe10	75
classe11	100

**Tabela 30:** Precisão considerando os descritores GCH e SIFT na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	3	1	0	0	0	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	2	0	0	2	0
classe5	0	0	0	0	0	0	4

**Tabela 31:** Matriz de confusão considerando os descritores GCH e SIFT na BC 2

## EXPERIMENTO 7 - DCD E DFC

O objetivo desse experimento foi avaliar a composição dos descritores de cor e forma, respectivamente, DCD e DFC. As Tabelas 32 e 33 mostram que a composição elevou o desempenho desses descritores isolados, conduzindo à uma taxa de 100% de precisão em todas as classes.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe10	100
classe11	100

**Tabela 32:** Precisão considerando os descritores DCD e DFC na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	4	0	0	0	0	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	0	0	0	4	0
classe5	0	0	0	0	0	0	4

**Tabela 33:** Matriz de confusão considerando os descritores DCD e DFC na BC 2

#### EXPERIMENTO 8 - DCD E SIFT

Assim como foi o caso para o experimento anterior, essa combinação conduziu à uma taxa de 100% de precisão para todas as classes (veja Tabelas 34 e 35).

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe10	100
classe11	100

**Tabela 34:** Precisão considerando os descritores DCD e SIFT na BC 2

Classe	classe3	classe4	classe11	classe1	classe2	classe10	classe5
classe3	4	0	0	0	0	0	0
classe4	0	4	0	0	0	0	0
classe11	0	0	5	0	0	0	0
classe1	0	0	0	3	0	0	0
classe2	0	0	0	0	5	0	0
classe10	0	0	0	0	0	4	0
classe5	0	0	0	0	0	0	4

**Tabela 35:** Matriz de confusão considerando os descritores DCD e SIFT na BC 2



### 5.2.3 AVALIAÇÃO DOS DESCRITORES PARA CLASSIFICAÇÃO DA BC 3

Como detalhado anteriormente, os experimentos com a BC 3 visam avaliar o desempenho de cada combinação de descritores em um processo de classificação supervisionada em uma base cujas imagens apresentam características distintas de cor e semelhantes de forma.

#### EXPERIMENTO 1 - GCH

Com a introdução de mais classes com cores próximas, esse descritor passa a ter um desempenho pior, como pode ser observado na Tabela 20. Devido às características do GCH, os erros (observados na Tabela 21) devem-se à proximidade das tonalidades das cores das amostras das classes.

Classe	Precisão (%)
classe1	66
classe2	100
classe3	50
classe4	25
classe5	100
classe6	50
classe7	25
classe8	100
classe9	75

**Tabela 36:** Precisão considerando o descritor GCH na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	2	0	0	0	0	0	0	2	0
classe9	0	3	1	0	0	0	0	0	0
classe4	0	2	1	1	0	0	0	0	0
classe7	0	0	0	1	1	0	0	2	0
classe1	0	0	0	1	2	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	1	0	0	0	0	1	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	4

**Tabela 37:** Matriz de confusão do experimento utilizando GCH na BC 3

#### EXPERIMENTO 2 - DCD

Com a adição de novas classes de flores, o descritor teve seu desempenho reduzido para várias classes, conforme mostra a Tabela 38. Isso ocorre porque as imagens adicionadas têm tonalidades próximas e há confusão na classificação, conforme pode-se observar na Tabela 39.

Classe	Precisão (%)
classe1	0
classe2	100
classe3	100
classe4	25
classe5	75
classe6	50
classe7	75
classe8	100
classe9	100

**Tabela 38:** Precisão considerando o descritor DCD na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0
classe9	0	4	0	0	0	0	0	0	0
classe4	0	2	1	0	0	0	0	1	0
classe7	0	1	0	3	0	0	0	0	0
classe1	0	2	0	1	0	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	0	1	0	0	0	1	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	1	0	0	3

**Tabela 39:** Matriz de confusão do experimento utilizando DCD na BC 3

Note que as classes 2 e 5 obtiveram uma taxa de 100% de precisão, devido ao fato de serem as únicas com cores dominantes amarela e branca, respectivamente. Essas classes também não foram associadas a nenhuma outra, como pode-se observar na matriz de confusão. A classe 8, por sua vez, embora também tenha obtido 100% de precisão, foi atribuída às classes 3 e 7. Este fato indica que as características destas classes são semelhantes, podendo em outros experimentos afetar o desempenho da classe 8.

A matriz de confusão mostra claramente alguns erros de classificação esperados como, por exemplo, entre as classes 4 e 9 ou 1 e 7. Tais erros enfatizam a necessidade de um descritor complementar, que possibilite diferenciar classes que de fato possuem cores semelhantes entre si.

Em comparação ao GCH, o DCD apresentou desempenho inferior apenas para as classes 1 e 5.

### EXPERIMENTO 3 - SIFT

Esse experimento teve 100% de precisão para todas as classes, conforme ilustra a Tabela 24. Este experimento mostra que, para a BC 3, os pontos de interesse extraídos pelo SIFT foram suficientes para classificar as imagens da base de testes sem nenhum erro (veja Tabela 41).

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	100

**Tabela 40:** Precisão considerando o descritor SIFT na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0
classe9	0	4	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0
classe1	0	0	0	0	3	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	0	0	0	0	0	2	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	4

**Tabela 41:** Matriz de confusão do experimento utilizando SIFT na BC 3

### EXPERIMENTO 4 - DFC

O desempenho da classificação ao considerar o DFC é ilustrado na Tabela 42. Assim como foi o caso para a BC 2, o desempenho do DFC foi fortemente influenciado pela sua limitação em representar determinados detalhes da forma. Portanto, uma flor sem pétalas espaçadas observada em uma determinada perspectiva pode gerar um descritor semelhança àquele de uma flor com pétalas espaçadas e pontiagudas.

Este problema pode ser evidenciado por erros detalhados na matriz de confusão. Observe que mesmo a classe 1, que possui uma forma diferenciada das demais por ser pontiaguda, apresenta uma precisão de apenas 33% por ter amostras classificadas como pertencentes às classes 3 e 4, que não possuem tal características. O melhor desempenho foi apresentado pela

classe 5, que possui pétalas bem separadas em comparação às demais classes, levando a uma melhor caracterização pelo DFC.

Classe	Precisão (%)
classe1	33
classe2	80
classe3	50
classe4	25
classe5	75
classe6	25
classe7	25
classe8	60
classe9	25

**Tabela 42:** Precisão considerando o descritor DFC na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	2	0	2	0	0	0	0	0	0
classe9	1	1	1	0	0	0	0	0	1
classe4	0	2	1	0	0	0	0	1	0
classe7	0	1	0	1	0	1	0	1	0
classe1	1	0	1	0	1	0	0	0	0
classe2	0	0	0	0	0	4	0	1	0
classe6	0	0	0	0	0	0	1	0	1
classe8	0	1	0	0	0	1	0	3	0
classe5	0	1	0	0	0	0	0	0	3

**Tabela 43:** Matriz de confusão do experimento utilizando DFC na BC 3

#### EXPERIMENTO 5 - GCH E DFC

Conforme ilustram as Tabelas 44 e 45, esta composição de descritores seguiu o padrão do GCH isolado (Tabela 36). Embora o DFC não tenha comprometido o GCH, mostrou-se desnecessário por não melhorar o desempenho em nenhum dos casos.

#### EXPERIMENTO 6 - GCH E SIFT

As Tabelas 46 e 47 mostram os resultados para a combinação GCH + SIFT, a qual não conduziu a nenhum erro de classificação. Contudo, como o SIFT obteve este mesmo desempenho quando considerado isoladamente (Tabela 40), o uso do GCH mostra-se desnecessário neste contexto.

Classe	Precisão (%)
classe1	66
classe2	100
classe3	50
classe4	25
classe5	100
classe6	50
classe7	25
classe8	100
classe9	75

**Tabela 44:** Precisão considerando os descritores GCH e DFC na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	2	0	0	0	0	0	0	2	0
classe9	0	3	1	0	0	0	0	0	0
classe4	0	2	1	1	0	0	0	0	0
classe7	0	0	0	1	1	0	0	2	0
classe1	0	0	0	1	2	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	1	0	0	0	0	1	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	4

**Tabela 45:** Matriz de confusão do experimento para os descritores GCH e DFC na BC 3

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	100

**Tabela 46:** Precisão considerando os descritores GCH e SIFT na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0
classe9	0	4	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0
classe1	0	0	0	0	3	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	0	0	0	0	0	2	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	4

**Tabela 47:** Matriz de confusão do experimento para os descritores GCH e SIFT na BC 3

## EXPERIMENTO 7 - DCD E DFC

Os resultados para a combinação DCD e DFC são apresentados nas Tabelas 48 e 49. O desempenho de tal composição mostrou-se equivalente àquele obtido para o descritor DCD

isoladamente (Tabela 38), exceto para a classe 1. Neste caso, o descritor DFC contribuiu para melhorar o desempenho desta classe (de 0% para 33%).

Classe	Precisão (%)
classe1	33
classe2	100
classe3	100
classe4	25
classe5	75
classe6	50
classe7	75
classe8	100
classe9	100

**Tabela 48:** Precisão considerando os descritores DCD e DFC na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0
classe9	0	4	0	0	0	0	0	0	0
classe4	0	2	1	0	0	0	0	1	0
classe7	0	1	0	3	0	0	0	0	0
classe1	0	2	0	0	1	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	0	1	0	0	0	1	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	1	0	0	3

**Tabela 49:** Matriz de confusão do experimento para os descritores DCD e DFC na BC 3

## EXPERIMENTO 8 - DCD E SIFT

Esta combinação não conduziu a nenhum erro de classificação (Tabelas 50 e 51). Contudo, assim como para o Experimento 6, como o SIFT obteve este mesmo desempenho quando considerado isoladamente (Tabela 40), o uso do DCD mostra-se desnecessário neste contexto.

### 5.2.4 AVALIAÇÃO DOS DESCRITORES PARA CLASSIFICAÇÃO DA BC 4

Como detalhado anteriormente, os experimentos com a BC 4 visam avaliar o desempenho de cada combinação de descritores em um processo de classificação supervisionada em uma base cujas imagens apresentam características distintas de cor e também de forma. Esta base representa, portanto, uma melhor aproximação de um cenário real.

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	100

**Tabela 50:** Precisão considerando os descritores DCD e SIFT na BC 3

Classe	classe3	classe9	classe4	classe7	classe1	classe2	classe6	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0
classe9	0	4	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0
classe1	0	0	0	0	3	0	0	0	0
classe2	0	0	0	0	0	5	0	0	0
classe6	0	0	0	0	0	0	2	0	0
classe8	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	4

**Tabela 51:** Matriz de confusão do experimento para os descritores DCD e SIFT na BC 3

## EXPERIMENTO 1 - GCH

Como esperado, para o GCH, as classes que contém tonalidades diferenciadas são as que apresentam melhor desempenho, como pode ser visto na Tabela 52. Em contrapartida as que compartilham tonalidades próximas com outras classes são afetadas, conforme detalha a matriz de confusão ilustrada na Tabela 53.

A classe 4 foi a que apresentou pior desempenho, tendo amostras atribuídas às classes 1, 9 e 10. As classes 10 e 11, por sua vez, que apresentam cores mais diferenciadas, obtiveram um bom desempenho.

Classe	Precisão (%)
classe1	100
classe2	80
classe3	75
classe4	25
classe5	75
classe6	100
classe7	50
classe8	100
classe9	50
classe10	100
classe11	100
classe12	75

**Tabela 52:** Precisão considerando o descritor GCH na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	3	0	0	0	0	0	0	0	0	0	1	0
classe9	0	2	1	0	0	1	0	0	0	0	0	0
classe4	0	1	1	0	0	1	0	0	0	1	0	0
classe7	0	0	0	2	0	0	0	0	0	0	2	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	0	0	0	0	3	0	0	0	0	0	0
classe12	0	0	0	0	0	0	3	1	0	0	0	0
classe2	0	0	0	0	0	0	1	4	0	0	0	0
classe6	0	0	0	0	0	0	0	0	2	0	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	1	0	0	0	0	0	0	3

**Tabela 53:** Matriz de confusão do experimento para o descritor GCH na BC 4

### EXPERIMENTO 2 - DCD

Os resultados para o descritor DCD são apresentados nas Tabelas 54 e 55. Como esperado, classes com cores semelhantes tem seu desempenho afetado. Em comparação ao GCH, o DCD apresenta desempenho inferior apenas para as classes 1 e 6.

Classe	Precisão (%)
classe1	66
classe2	100
classe3	100
classe4	50
classe5	75
classe6	0
classe7	75
classe8	100
classe9	75
classe10	100
classe11	100
classe12	100

**Tabela 54:** Precisão considerando o descritor DCD na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0	0	0	0
classe9	0	3	0	0	0	0	0	0	0	1	0	0
classe4	0	1	2	0	0	0	0	0	0	0	1	0
classe7	0	0	0	3	0	0	0	0	0	0	1	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	1	0	0	0	2	0	0	0	0	0	0
classe12	0	0	0	0	0	0	4	0	0	0	0	0
classe2	0	0	0	0	0	0	0	5	0	0	0	0
classe6	0	1	0	0	0	0	0	0	0	1	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	1	0	0	0	0	0	0	3

**Tabela 55:** Matriz de confusão do experimento para o descritor DCD na BC 4

### EXPERIMENTO 3 - SIFT

As Tabelas 56 e 57 mostram o desempenho do descritor SIFT na classificação da BC 4. A classe que apresentou o pior desempenho foi a 2, o que se deve à inclusão da classe 12, a qual



possui características de cor e forma semelhantes. Isso mostra a necessidade, já enfatizada, de utilizar um vetor de características composto por informações que consideram diferentes características.

Classe	Precisão (%)
classe1	100
classe2	20
classe3	75
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	75
classe10	50
classe11	100
classe12	75

**Tabela 56:** Precisão considerando o descritor SIFT na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	3	0	0	0	1	0	0	0	0	0	0	0
classe9	0	3	0	0	1	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0	0	0	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	0	0	0	0	3	0	0	0	0	0	0
classe12	0	0	0	0	1	0	3	0	0	0	0	0
classe2	0	0	0	0	4	0	0	1	0	0	0	0
classe6	0	0	0	0	0	0	0	0	2	0	0	0
classe10	0	0	0	0	2	0	0	0	0	2	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	0	0	0	4

**Tabela 57:** Matriz de confusão do experimento para o descritor SIFT na BC 4

#### EXPERIMENTO 4 - DFC

O desempenho do DFC é mostrado nas Tabelas 58 e 59. O baixo desempenho deve-se aos fatores já amplamente discutidos nas bases anteriores.

Classe	Precisão (%)
classe1	66
classe2	25
classe3	25
classe4	25
classe5	75
classe6	25
classe7	25
classe8	60
classe9	25
classe10	50
classe11	20
classe12	25

**Tabela 58:** Precisão considerando o descritor DFC na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	1	1	0	0	0	1	0	0	0	1	0	0
classe9	1	1	1	0	0	0	0	0	0	0	0	1
classe4	1	1	1	0	1	0	0	0	0	0	0	0
classe7	1	0	0	1	0	1	0	0	0	0	1	0
classe11	0	2	0	0	1	0	0	0	0	1	0	1
classe1	0	0	0	0	0	2	0	0	0	0	1	0
classe12	0	0	1	0	0	0	1	0	0	2	0	0
classe2	0	0	0	0	0	0	4	1	0	0	0	0
classe6	0	1	0	0	0	1	0	0	1	0	1	0
classe10	1	0	0	0	0	1	0	0	0	2	0	0
classe8	0	0	0	0	0	0	1	1	0	0	3	0
classe5	0	0	1	0	0	0	0	0	0	0	0	3

**Tabela 59:** Matriz de confusão do experimento utilizando DFC na BC 4

## EXPERIMENTO 5 - GCH E DFC

Os resultados descritos nas Tabelas 60 e 61 mostram que a composição conduz à resultados equivalentes aos do descritor GCH separadamente (Tabela 52). Portanto, o uso do DFC neste contexto mostra-se desnecessário.

Classe	Precisão (%)
classe1	100
classe2	80
classe3	75
classe4	25
classe5	75
classe6	100
classe7	50
classe8	100
classe9	50
classe10	100
classe11	100
classe12	75

**Tabela 60:** Precisão considerando os descritores GCH e DFC na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	3	0	0	0	0	0	0	0	0	0	1	0
classe9	0	2	1	0	0	1	0	0	0	0	0	0
classe4	0	1	1	0	0	1	0	0	0	1	0	0
classe7	0	0	0	2	0	0	0	0	0	0	2	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	0	0	0	0	3	0	0	0	0	0	0
classe12	0	0	0	0	0	0	3	1	0	0	0	0
classe2	0	0	0	0	0	0	1	4	0	0	0	0
classe6	0	0	0	0	0	0	0	0	2	0	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	1	0	0	0	0	0	0	3

**Tabela 61:** Matriz de confusão do experimento para os descritores GCH e DFC na BC 4

## EXPERIMENTO 6 - GCH E SIFT

O objetivo desse experimento foi avaliar a composição dos descritores GCH e SIFT como forma de verificar o desempenho na classificação nessa base. Os resultados descritos nas Tabelas 62 e 63 mostram que os resultados são equivalentes àqueles obtidos com o SIFT separadamente (Tabela 56), exceto para as classes 2 e 10, casos em que o CGH mostrou-se uma importante característica complementar.

Classe	Precisão (%)
classe1	100
classe2	80
classe3	75
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	75
classe10	100
classe11	100
classe12	75

**Tabela 62:** Precisão considerando os descritores GCH e SIFT na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	3	0	0	0	1	0	0	0	0	0	0	0
classe9	0	3	0	0	1	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0	0	0	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	0	0	0	0	3	0	0	0	0	0	0
classe12	0	0	0	0	1	0	3	0	0	0	0	0
classe2	0	0	0	0	1	0	0	4	0	0	0	0
classe6	0	0	0	0	0	0	0	0	2	0	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	0	0	0	4

**Tabela 63:** Matriz de confusão do experimento para os descritores GCH e SIFT na BC 4

## EXPERIMENTO 7 - DCD E DFC

O resultado da composição dos descritores DCD e DFC é ilustrado nas Tabelas 64 e 65. Pode-se observar que tal composição conduz à resultados equivalentes aos do descritor DCD isolado (Tabela 54). Portanto, o uso do DFC neste contexto mostra-se desnecessário, assim como foi o caso para o Experimento 5.

Classe	Precisão (%)
classe1	66
classe2	100
classe3	100
classe4	50
classe5	75
classe6	0
classe7	75
classe8	100
classe9	75
classe10	100
classe11	100
classe12	100

**Tabela 64:** Precisão considerando os descritores DCD e DFC na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0	0	0	0
classe9	0	3	0	0	0	0	0	0	0	1	0	0
classe4	0	1	2	0	0	0	0	0	0	0	1	0
classe7	0	0	0	3	0	0	0	0	0	0	1	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	1	0	0	0	2	0	0	0	0	0	0
classe12	0	0	0	0	0	0	4	0	0	0	0	0
classe2	0	0	0	0	0	0	0	5	0	0	0	0
classe6	0	1	0	0	0	0	0	0	0	1	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	1	0	0	0	0	0	0	3

**Tabela 65:** Matriz de confusão do experimento para os descritores DCD e DFC na BC 4

## EXPERIMENTO 8 - DCD E SIFT

As Tabelas 66 e Tabela 67 ilustram os resultados para a composição DCD e SIFT, a qual conduziu à taxa de 100% de precisão para todas as classes exceto a 9 (que teve uma das amostras classificadas como pertencente à classe 11).

Classe	Precisão (%)
classe1	100
classe2	100
classe3	100
classe4	100
classe5	100
classe6	100
classe7	100
classe8	100
classe9	75
classe10	100
classe11	100
classe12	100

**Tabela 66:** Precisão considerando os descritores DCD e SIFT na BC 4

Classe	classe3	classe9	classe4	classe7	classe11	classe1	classe12	classe2	classe6	classe10	classe8	classe5
classe3	4	0	0	0	0	0	0	0	0	0	0	0
classe9	0	3	0	0	1	0	0	0	0	0	0	0
classe4	0	0	4	0	0	0	0	0	0	0	0	0
classe7	0	0	0	4	0	0	0	0	0	0	0	0
classe11	0	0	0	0	5	0	0	0	0	0	0	0
classe1	0	0	0	0	0	3	0	0	0	0	0	0
classe12	0	0	0	0	0	0	4	0	0	0	0	0
classe2	0	0	0	0	0	0	0	5	0	0	0	0
classe6	0	0	0	0	0	0	0	0	2	0	0	0
classe10	0	0	0	0	0	0	0	0	0	4	0	0
classe8	0	0	0	0	0	0	0	0	0	0	5	0
classe5	0	0	0	0	0	0	0	0	0	0	0	4

**Tabela 67:** Matriz de confusão do experimento para os descritores DCD e SIFT na BC 4

### 5.2.5 CONSIDERAÇÕES GERAIS

Como conclusões gerais, pode-se afirmar que o descritor de cor DCD apresentou desempenho superior ao GCH para a grande maioria dos casos. Isso se deve em parte à característica da base de imagens sendo considerada, a qual é composta por imagens de flores que possuem uma cor dominante.

O descritor de forma DFC, por sua vez, apresentou um baixo desempenho. Como o feixo convexo é representado como um “contorno” de pontos, há flores cuja forma acaba representada por uma aproximação da forma original, onde podem existir características de curvatura mais específicas. O SIFT, por sua vez, apresentou um ótimo desempenho.

Em alguns experimentos, a combinação de descritores apresentou desempenho equivalente ao uso de um descritor isolado. Isso se deve tanto às características controladas das bases (se as formas já são distintas, um descritor de forma tende a não contribuir para a classificação) quando às limitações da representação dos descritores (como é o caso do DFC). As melhores combinações de descritores foram DCD+SIFT e GCH+SIFT.

A classe 4 foi a que apresentou o pior desempenho para os descritores de cor, o que é justificado pelo fato de as amostras possuírem cores semelhantes às de outras classes. As classes 8, 10 e 11 apresentaram o melhor desempenho neste contexto, devido às características bem definidas de cor que apresentam. O desempenho para o descritor de forma SIFT foi afetado com a inclusão de formas parecidas.

A próxima seção descreve o processo de recuperação de imagens propriamente dito.

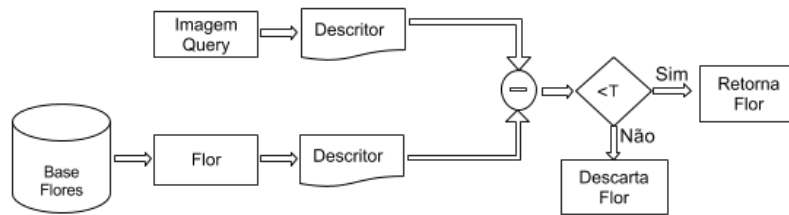
### 5.3 CBIR

Visando uma recuperação de imagens que seja tanto robusta quanto eficiente, foi realizado um estudo da precisão associada aos descritores em um processo de classificação supervisionada. Isso se deve ao fato da abordagem CBIR estar relacionada com este conceito: em teoria, as imagens recuperadas são aquelas classificadas como pertencentes à mesma classe da imagem *query*.

Com base nos testes experimentais, foi possível observar a importância de se considerar um vetor de características contendo descritores que considerem diferentes características (cor, forma e pontos de interesse, no contexto deste trabalho). Este fato é crucial para a robustez do sistema. Neste contexto, para a avaliação do processo de CBIR foi utilizada a combinação de descritores DCD e SIFT, que apresentou os melhores resultados na avaliação da seção anterior.

O processo de recuperação de imagens segue a metodologia ilustrada na Figura 37. Em suma, o vetor de características de cada imagem *query* é comparado com aqueles correspondentes às imagens armazenadas na base de busca, composta pelas imagens da BC 4, discutida anteriormente. Tal comparação mede a similaridade entre os vetores com base na análise da distância L2. Caso ela seja menor que um limiar  $T$ , a imagem correspondente da base é retornada. Caso contrário, é descartada.

O caso de uso Executar Experimento CBIR, descrito na Seção 6.2.3.2, explica de modo breve o processo envolvendo a execução do experimento do ponto de vista do usuário. Resumidamente, o fluxo básico é composto pelos seguintes passos:



**Figura 37: Metodologia CBIR**

1. o usuário solicita ao sistema executar um experimento CBIR;
2. o sistema pede ao usuário o caminho do arquivo de modelo de experimento;
3. o sistema carrega o modelo de experimento em suas entidades;
4. o sistema carrega os vetores de características associados às imagens da base de busca;
5. o sistema seleciona as imagens *query* e extrai os respectivos vetores de características;
6. Para cada imagem *query*, o sistema extrai da base aquelas cujo vetor de características tem distância menor que o limiar especificado,  $T$ , e salva os resultados. O valor de  $T$  é discutido abaixo;
7. o sistema calcula a precisão e revocação para cada classe de entrada e depois para todo o experimento;
8. o sistema salva os dados de precisão e revocação em disco.

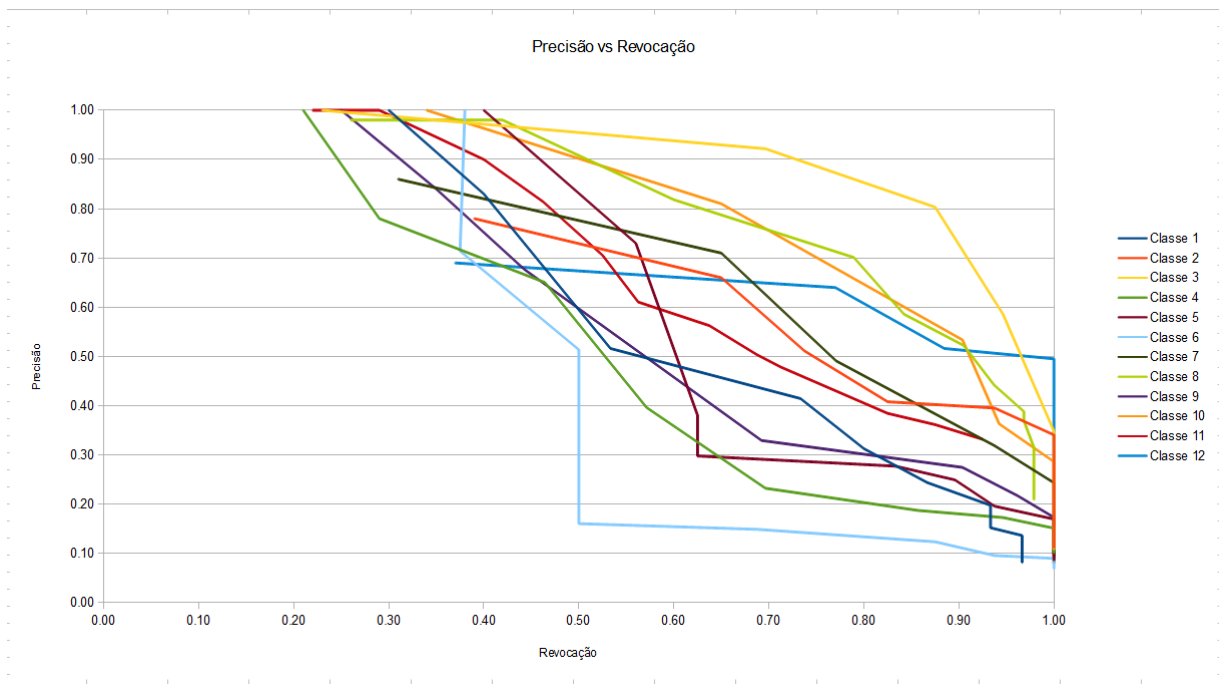
### 5.3.1 DISCUSSÃO DOS RESULTADOS

Como já mencionado, foram utilizados os descritores DCD e SIFT, que apresentaram os melhores resultados na avaliação de descritores. Com a variação do valor do limiar é possível calcular a curva precisão  $\times$  revocação para cada classe, segundo a abordagem proposta em (SMITH; CHANG, 1996). Desta forma, pode-se comparar o desempenho de cada classe mediante a variação limiar.

Para visualizar o impacto da escolha do limiar de um experimento por classe de imagens, elaborou-se o seguinte estudo. Considerando o intervalo  $T$  e a normalização das saídas dos descritores SIFT e DCD para um valor entre 0 e 1, variou-se o limiar  $T$  em intervalos  $\delta = 0,01$  entre  $T = 0,2$  e  $T=0,3$ . Para cada um dos valores, executou-se um experimento.

Para cada um dos experimentos, a média da precisão e revocação das *queries* de cada classe originou um ponto de precisão–revocação. Estes pontos foram apresentados sob a forma de uma curva precisão–revocação, ilustrada na Figura 38.





**Figura 38: Curva precisão versus revocação**

Para avaliar o desempenho de cada classe, e a partir da ideia da curva de precisão–revocação apresentada no capítulo de estudo teórico, analisaram–se as curvas a partir do seguinte critério de desempenho: seja a curva  $c_i$ , composta de um número discreto de pontos  $(p_i, r_i)$ , onde  $i$  é a classe de imagens. A curva  $c_i$  possui melhor desempenho se a distância média, obtida pela média aritmética das distâncias entre a origem e cada um de seus pontos é maior do que a distância média de outra curva qualquer  $c_j$ , desde que  $j \neq i$ .

Seguindo este critério, o melhor desempenho foi obtido pela classe 3, que tem como justificativa de que apresenta tonalidade laranja forte e forma com pétalas bem definidas. Analisando o conjunto de *queries*, as imagens desta classe são visualmente mais distintas das demais.

O segundo melhor desempenho foi obtido pela classe 8 a qual, assim como a classe 3, possui forma com pétalas bem definidas e redonda, porém com forte tom vermelho que a diferencia de forma significativa das outras da base.

Por outro lado, o pior desempenho foi obtido pela classe 6, que apresenta cor rosa muito similar a outras classes da base. Além disso, sua forma apresenta perturbações devido à algumas componentes extras da flor. A classe 4 teve o segundo pior desempenho, justificado pelo fato de também possuir uma cor similar à outras da base, ao mesmo tempo em que a forma das pétalas muda dependendo da amostra, prejudicando o desempenho do descritor de forma.

As outras classes tiveram bom desempenho e seguem o padrão decrescente em relação à precisão à medida que a revocação aumenta, como esperado. As classes 2 e 12 apresentaram

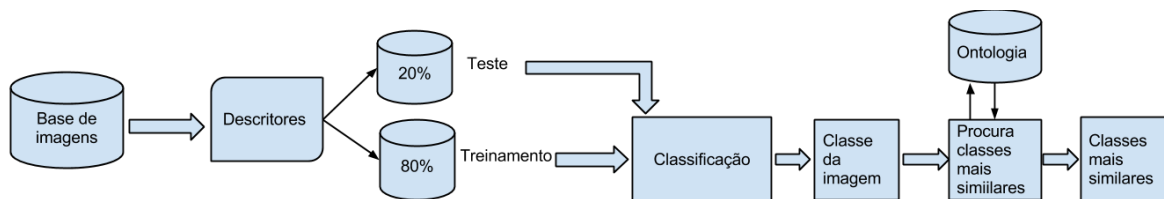
uma precisão menor quando a revocação é baixa devido à característica arredondada da forma, pétalas agulhadas e cor amarela, presente nas duas classes.

A classe 7 apresentou boa precisão mesmo com o aumento da revocação, o que está associado à cor vermelha da flor, forma homogênea arredondada com pouca variação e pétalas bem definidas. A classe 5 teve uma queda acentuada na precisão para revocação acima de 50%, justificado pela presença de outros tons junto à cor dominante branca. Além disso, algumas componentes da flor influenciam o desempenho do descritor de forma.

Para as classes 9, 10 e 11, a queda é mais suave, o que é ocasionado pelas características menos comuns na base de imagem, como a forma das três e a cor. Contudo, com o aumento da revocação ainda é possível observar uma queda considerável da precisão.

#### 5.4 METODOLOGIA PARA USO DE ONTOLOGIAS COMO APOIO AO UM SISTEMA DE RECUPERAÇÃO DE IMAGENS

A metodologia proposta para utilização de ontologias como apoio a um sistema de recuperação de imagens baseado em conteúdo é apresentada na Figura 39.



**Figura 39: Metodologia de sistema de recuperação de imagens com ontologia adotada**

O procedimento se dá da seguinte forma. A partir da base de imagens, calculam-se os descritores, obtendo-se os vetores de características. Destes, 20% são selecionados para execução de *queries* (teste) e os 80% restantes são utilizados como treinamento. A execução de cada *query* resulta na atribuição de uma classe à ela. Com auxílio da ontologia, procuram-se classes mais similares para a classe encontrada no passo anterior e finalmente tem-se como resultado todas as imagens pertencentes às classes encontradas.

Portanto, o uso de ontologias pode ser visto como uma extensão da etapa de validação, onde utiliza-se uma ontologia como artefato para identificar objetos similares. Para facilitar a organização ela é dividida em duas etapas: construção da ontologia (segundo o processo descrito na Seção 3.2) e execução do experimento.

Assim como no experimento de CBIR, foi utilizada a Base Controlada 4 (BC4) para a

Classe	Nome real (em inglês)	Cor	Tamanho real (tronco)	Localização geográfica (nativa)
classe1	<i>Anthurium</i>	Branca, vermelha	Entre 15 e 30 cm	Américas central e do sul
classe2	<i>Colt's foot</i>	Amarela	Entre 10 e 30 cm	Europa e Ásia
classe3	<i>Watercress</i>	Laranja	Entre 0,3 e 0,6 m	Europa
classe4	<i>Hibiscus</i>	Vermelha, Branca	Entre 4 e 18 cm	Ásia, América central
classe5	<i>Ruby-Lipped Cattleya</i>	Roxa, Branca	Entre 15 a 20 cm	América do sul (Brasil)
classe6	<i>Ball Moss</i>	Roxa	Entre 5 e 15 cm	Américas (Norte, Central e Sul)
classe7	<i>Barbeton daisy</i>	Amarela, Laranja, Vermelha	Entre 6 e 10 cm	África do sul
classe8	<i>Bishop of Landalf</i>	Vermelha, Laranja, Roxa	Entre 1 e 2 m	América central
classe9	<i>Rose</i>	Branca, vermelha, rosa	De alguns centímetros até 7 metros	China
classe10	<i>Artichoke</i>	Roxa, verde	De 1,4 a 2 m	Região mediterrânea
classe11	<i>Giant white arum lily</i>	Branca	0,6 a 1 m	África do sul
classe12	<i>Dandelion</i>	Amarela	5 a 45 cm	Europa

**Tabela 68:** Informações sobre cada classe de objetos

formulação deste experimento. Portanto, quando a palavra “classe” for mencionada nas seções abaixo, considera-se como pertencente à BC4. A documentação do processo é dividida em três seções: especificação (5.4.1), conceitualização (5.4.2) e implementação (5.4.3).

#### 5.4.1 ESPECIFICAÇÃO DA ONTOLOGIA

Como mencionado, o objetivo da ontologia é auxiliar no processo de busca de imagens por conteúdo. Neste contexto, para uma dada classe de entrada, um método de busca  $P$  irá localizar quais classes são mais similares baseadas em um conjunto de informações, as quais devem ser complementares àquelas fornecidas pelos vetores de características.

Como as classes referem-se a espécies de flores, na ontologia proposta o relacionamento entre os conceitos se dará por três atributos: “cor”, “tamanho real” e “localização geográfica”. Definido o propósito e escopo da ontologia, é necessário representar, para cada classe de imagem, estas informações, que serão utilizadas para a conceitualização da ontologia. A Tabela 68 lista as informações do arquivo de especificação. Estas foram obtidas através da base de dados de plantas Floridata (FLORIDATA, 2013).

Embora a cor seja uma informação definida por descritores, foi observado que algumas espécies de flores apresentam cores distintas (o *Anthurium*, por exemplo, pode ser branco ou vermelho). O tamanho real é uma informação que não pode ser representada por descritores, pois a área de uma região de interesse depende da escala em que a imagem foi adquirida. Por exemplo, tanto um girassol quanto um dente-de-leão podem ter a mesma área em uma imagem, embora tenham tamanhos distintos. A localização geográfica é outra informação que só pode ser representada por metadados textuais.

#### 5.4.2 CONCEITUALIZAÇÃO

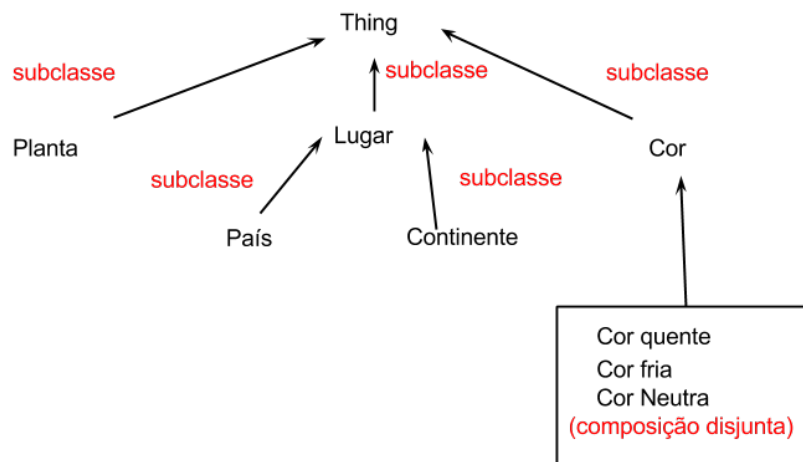
Baseado nas informações da Tabela 68, que traz a especificação da ontologia, ela será construída através do método *Methontology*, como apresentado na Seção 3.2.1. Este procedimento é dividido nas tarefas descritas abaixo.

##### 5.4.2.1 Glossário de termos

O objetivo desta tarefa é definir o glossário de termos da ontologia. O glossário de conceitos e instâncias é apresentado na Tabela 69. As relações são apresentadas na Tabela 70.

##### 5.4.2.2 Taxonomias de conceito

A taxonomia de conceito é apresentada na Figura 40. A classe *Cor* subdivide-se em uma composição disjunta, de conceitos *Cor neutra*, *Cor fria* e *Cor Quente*.



**Figura 40: Taxonomia de conceitos da ontologia**

Nome	Sinônimos	Acrônimos	Descrição	Tipo
<i>Planta</i>	–	–	Conceito planta	Conceito
<i>Cor</i>	–	–	Conceito cor	Conceito
<i>Cor Quente</i>	–	–	Conceito “cor quente”	Conceito
<i>Cor Fria</i>	–	–	Conceito “cor fria”	Conceito
<i>Cor Neutra</i>	–	–	Conceito “cor neutra”	Conceito
<i>Tamanho</i>	–	–	Conceito tamanho	Conceito
<i>Lugar</i>	–	–	Conceito lugar	Conceito
<i>Continente</i>	–	–	Conceito continente	Conceito
<i>País</i>	–	–	Conceito país	Conceito
<i>Anthurium</i>	–	–	Planta <i>Anthurium</i>	Instância
<i>Colt's foot</i>	–	–	Planta <i>Colt's foot</i>	Instância
<i>Watercress</i>	–	–	Planta <i>Watercress</i>	Instância
<i>Hibiscusm</i>	–	–	Planta <i>Hibiscus</i>	Instância
<i>Ruby-Lipped Cattleya</i>	–	–	Planta <i>Ruby-Lipped Cattleya</i>	Instância
<i>Ball Moss</i>	–	–	Planta <i>Ball Moss</i>	Instância
<i>Barbeton daisy</i>	–	–	Planta <i>Barbeton daisy</i>	Instância
<i>Bishop of Landalf</i>	–	–	Planta <i>Bishop of Landalf</i>	Instância
<i>Rose</i>	–	–	Planta <i>Rose</i>	Instância
<i>Artichoke</i>	–	–	Planta <i>Artichoke</i>	Instância
<i>Giant white arum lily</i>	–	–	Planta <i>Giant white arum lily</i>	Instância
<i>Dandelion</i>	–	–	Planta <i>Dandelion</i>	Instância
<i>Amarela</i>	–	–	Cor amarela	Instância
<i>Branca</i>	–	–	Cor branca	Instância
<i>Vermelha</i>	–	–	Cor vermelha	Instância
<i>Laranja</i>	–	–	Cor laranja	Instância
<i>Roxa</i>	–	–	Cor roxa	Instância
<i>Rosa</i>	–	–	Cor rosa	Instância
<i>Verde</i>	–	–	Cor verde	Instância
<i>Grande</i>	–	–	Tamanho grande	Instância
<i>Pequeno</i>	–	–	Tamanho pequeno	Instância
<i>Médio</i>	–	–	Tamanho médio	Instância
<i>América do Sul</i>	–	–	Continente América do Sul	Instância
<i>América Central</i>	–	–	Continente América Central	Instância
<i>América do Norte</i>	–	–	Continente América do Norte	Instância
<i>Europa</i>	–	–	Continente América do Norte	Instância
<i>Ásia</i>	–	–	Continente Ásia	Instância
<i>África</i>	–	–	Continente África	Instância
<i>Brasil</i>	–	–	País Brasil	Instância
<i>África do sul</i>	–	–	País África do Sul	Instância
<i>China</i>	–	–	País China	Instância

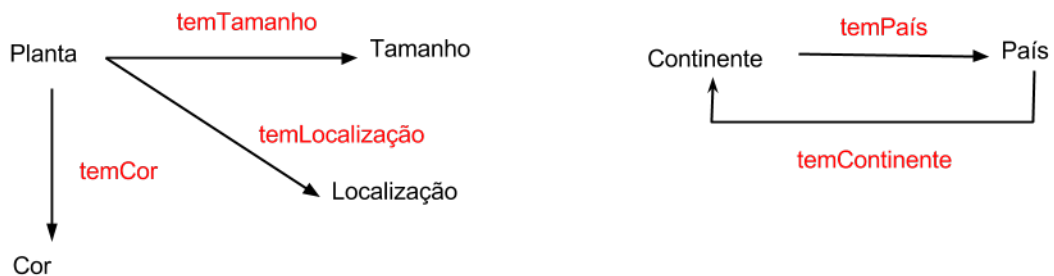
**Tabela 69:** Glossário de conceitos e instâncias da ontologia

#### 5.4.2.3 Diagrama de relações binárias *ad hoc*

O diagrama de relações binárias *ad hoc* é apresentado na Figura 41.

Nome	Sinônimos	Acrônimos	Descrição	Tipo
$temCor(p,c)$	–	–	Indica que a planta p possui a cor c	Relação
$temLocalização(p,l)$	–	–	Indica que a planta p possui a localização l	Relação
$temTamanho(p,t)$	–	–	Indica que a planta p possui o tamanho t	Relação

**Tabela 70:** Glossário de relações da ontologia



**Figura 41:** Diagrama de relações binárias ad hoc

#### 5.4.2.4 Dicionário de conceitos

O dicionário de conceitos é apresentado na Tabela 71.

#### 5.4.2.5 Definição detalhada das relações *ad hoc*

A definição detalhada das relações *ad hoc* é apresentada na Tabela 72.

#### 5.4.2.6 Definição das instâncias

A última etapa tem como objetivo definir as instâncias e qual o valor de cada uma de suas relações. Para o conceito “Planta”, as tabelas 73 e 74 os apresentam. Para os conceitos “País” e “Continente”, os valores são apresentados na tabela 75.

Nome do conceito	Instâncias	Atributos de classe	Atributos de instância	Relações
Planta	<i>Anthurium, Colt's foot, Watercress, Hibiscus, Ruby-Lipped Cattleya, Ball Moss, Barbeton daisy, Bishop of Landalf, Rose, Artichoke, Giant white arum lily, Dandelion</i>	–	–	temTamanho, temLocalização, temCor
Cor	Amarela, Branca, Vermelha, Laranja, Roxa, Rosa, Verde	–	–	–
Tamanho	Grande, Médio, Pequeno	–	–	–
Lugar	–	–	–	–
Continente	América do Sul, América Central, América do Norte, Europa, Ásia	–	–	temPaís
País	Brasil, África do sul, China	–	–	temContinente

**Tabela 71:** Dicionário de conceitos

Nome da relação	Conceito de origem	Cardinalidade	Conceito destino	Relação inversa
temTamanho	Planta	1	Tamanho	–
temLocalização	Planta	1 ... N	Lugar	–
temCor	Planta	1 ... N	Cor	–
temPaís	Continente	1..N	País	temContinente
temContinente	País	1	Continente	temPaís

**Tabela 72:** Definição das relações *ad hoc*

#### 5.4.3 IMPLEMENTAÇÃO E BUSCAS NA ONTOLOGIA

A implementação da ontologia foi realizada com auxílio do *software Protégé* (RESEARCH, 2011). Utilizou-se a linguagem *OWL-DL*. A Figura 42 apresenta a visualização da ontologia implementada no *Protégé*.

Nome da instância	Conceito base	Nome da relação	Instância relativa	Conceito da instância relativa
<i>Anthurium</i>	Planta	temCor temTamanho temLocalização	Vermelha Pequeno América Central, América do Sul	Cor Tamanho Continente
<i>Colt's foot</i>	Planta	temCor temTamanho temLocalização	Amarela Pequeno Europa, Ásia	Cor Tamanho Continente
<i>Watercress</i>	Planta	temCor temTamanho temLocalização	Laranja Médio Europa	Cor Tamanho Continente
<i>Hibiscus</i>	Planta	temCor  temTamanho temLocalização	Vermelha, Branca Pequeno Ásia, amé- rica central	Cor  Tamanho Continente
<i>Ruby-Lipped Cattleya</i>	Planta	temCor  temTamanho temLocalização	Roxa, Branca Pequeno Brasil	Cor  Tamanho País
<i>Ball Moss</i>	Planta	temCor temTamanho temLocalização	Roxa Pequeno América do Norte, América Central, América do Sul	Cor Tamanho Continente
<i>Barbeton daisy</i>	Planta	temCor  temTamanho temLocalização	Amarela, Laranja, Vermelha Pequeno África do Sul	Cor  Tamanho País

**Tabela 73:** Definição das instâncias do conceito “Planta”, parte 1

Com base na metodologia proposta, o mecanismo de buscas seria realizado da forma ilustrada na Figura 43.

Em resumo, as imagens seriam classificadas utilizando-se os mesmos descritores empregados no experimento de CBIR (DCD e SIFT, no contexto deste trabalho) por meio de um



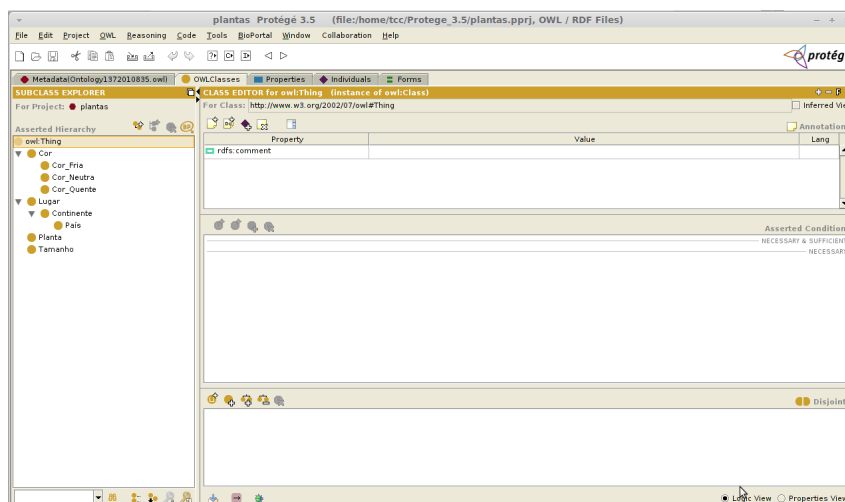


Figura 42: Representação da ontologia no *Protégé*

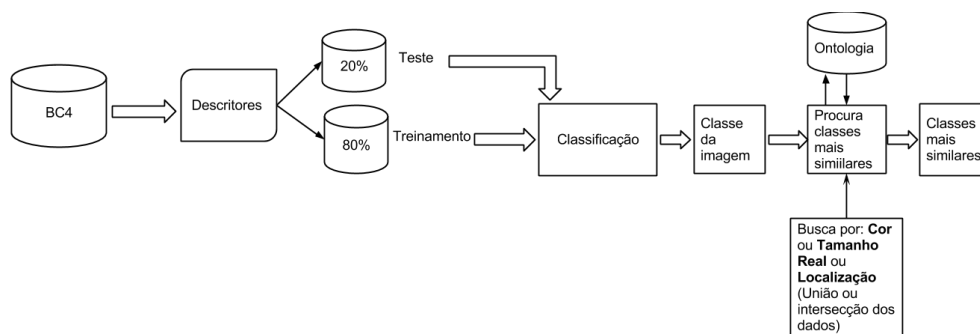


Figura 43: Metodologia de buscas adotadas para a ontologia

Nome da instância	Conceito base	Nome da relação	Instância relativa	Conceito da instância relativa
<i>Bishop of Landalf</i>	Planta	temCor temTamanho temLocalização	Vermelha Grande América Central	Cor Tamanho Continente
<i>Rose</i>	Planta	temCor  temTamanho temLocalização	Branca, Vermelha, Rosa Grande China	Cor  Tamanho País
<i>Artichoke</i>	Planta	temCor  temTamanho temLocalização	Roxa, Verde Grande Europa	Cor  Tamanho Continente
<i>Giant white arum lily</i>	Planta	temCor  temTamanho temLocalização	Roxa, Verde Médio África do Sul	Cor  Tamanho País
<i>Dandelion</i>	Planta	temCor temTamanho temLocalização	Amarela Médio Europa	Cor Tamanho Continente

**Tabela 74:** Definição das instâncias do conceito “Planta”, parte 2

Nome da instância	Conceito base	Nome da relação	Instância relativa	Conceito da instância relativa
Brasil	País	temContinente	América do Sul	Continente
África do Sul	País	temContinente	África	Continente
China	País	temContinente	Ásia	Continente
América do Sul	País	temPaís	Brasil	País
África	País	temPaís	África do Sul	País
Ásia	País	temPaís	China	País

**Tabela 75:** Definição das instâncias dos conceitos País e Continente

processo supervisionado considerando o KNN. Este processo é composto pelos passos apresentados no Capítulo 4 e tem como resultado a classe à qual pertence o objeto sendo buscado.

Dispondo da classe do objeto e da ontologia construída, a busca por imagens semelhantes seria realizada com auxílio da ontologia. O usuário do experimento escolhe um critério de busca (“cor”, “tamanho real”, ou “localização geográfica”), podendo-os assumir valores iguais

(mesmo valor das propriedades) ou semelhantes (sendo necessário definir um critério de semelhança para as classes).

A busca por mais de um parâmetro, por exemplo “cor” e “tamanho real” iguais, pode ser vista como duas buscas distintas, sendo uma apenas de “cor” e a outra apenas de “tamanho real”, sendo os resultados agrupados. Uma proposta é fornecer a possibilidade de escolher entre as operações de conjuntos de união e intersecção para fazê-lo.

Este processo está propenso a erros, tanto por questões de limitação dos descritores quando pela presença de amostras com características similares àquelas de outras classes. Por exemplo, este resultado é também analisado com base na ontologia criada. Neste ponto, poderia ser possível selecionar quais os atributos deseja-se considerar.

## 5.5 PROPOSTA DE USO DA ONTOLOGIA COMO APOIO AO PROCESSO DE RECUPERAÇÃO

Na Seção 5.3.1, foram discutidos os resultados obtidos no processo de recuperação de imagens baseado em descritores. Com base na análise do gráfico precisão  $\times$  revocação, foi possível observar o baixo desempenho geral para classificação de amostras das classes 4 e 6. Isso se deve principalmente ao fato de tais classes possuírem cores relativamente similares às de outras classes, o que influencia os descritores.

Contudo, observando-se as Tabelas 73 e 74, pode-se observar que a classe 4 tem como atributos “tamanho real” e “localização geográfica” os valores “pequeno” e “Ásia e América Central”. Embora tais atributos ocorram também nas classes 2 e 6, estas possuem as cores amarela e roxa, respectivamente, as quais são facilmente distinguidas da cor vermelha (presente na classe 4) pelo descritor DCD. Por esta razão, os erros resultantes da recuperação de imagens da classe 4 utilizando descritores possivelmente não se devem à associação às classes 2 e 6. Portanto, tais erros poderiam ser minimizados pela inclusão de uma etapa adicional que considerasse os atributos “tamanho real” e “localização geográfica”.

O mesmo raciocínio se aplica aos erros de classificação da classe 6, que também apresentou baixo desempenho na recuperação. Ela é única amostra a apresentar um tamanho pequeno, a cor roxa e estar localizada no continente americano.

Com base na análise das instâncias do conceito “Planta”, Tabelas 73 e 74, pode-se observar que as taxas de precisão e revocação das classes 9, 10 e 11 também podem ser melhoradas explorando-se os atributos “tamanho real” e/ou “localização geográfica”. A classe 11 é a única de tamanho médio localizado na África do Sul. Da mesma forma, as classes 9 e 10 são as únicas

Classe	Nome real (em inglês)	Cor	Tamanho real (tronco)	Localização geográfica (nativa)
classe1	<i>Anthurium</i>	Branca, vermelha	Entre 15 e 30 cm	Américas central e do sul
classe5	<i>Ruby-Lipped Cattleya</i>	Roxa, Branca	Entre 15 a 20 cm	América do sul (Brasil)

Classe	Nome real (em inglês)	Cor	Tamanho real (tronco)	Localização geográfica (nativa)
classe2	<i>Colt's foot</i>	Amarela	Entre 10 e 30 cm	Europa e Ásia
classe12	<i>Dandelion</i>	Amarela	5 a 45 cm	Europa

**Tabela 76:** Informações sobre cada classe de objetos

com tamanho grande localizadas na China e Europa, respectivamente.

O desempenho da classe 5, que na recuperação baseada em descritores teve uma queda acentuada na precisão para revocação acima de 50% , também pode ser melhorado se o atributo “localização geográfica” for considerado na etapa de apoio baseada em ontologias. A classe 5 é a única em que este atributo está associado exclusivamente ao Brasil.

Assim como o uso de uma abordagem baseada exclusivamente na análise dos descritores apresenta limitações, a utilização exclusiva de ontologias também apresenta problemas. Além daqueles já mencionados, associados à fatores que incluem a dificuldade na especificação de uma ontologia e à subjetividade no processo de rotulação, ambiguidades ainda podem estar presentes. Neste trabalho, foram consideradas apenas amostras vermelhas na classe 1. Contudo, tal flor também pode ser branca, como descrito na Tabela 76 (extraída da Tabela 68). Neste caso, se apenas os atributos da ontologia fossem considerados, uma busca pelos termos “branca”, “pequena”, “América so Sul” poderia retornar tanto amostras da classe 1 quanto da classe 5.

Observe que os atributos “tamanho real” e “localização geográfica” não podem ser representados por meio de medidas extraídas diretamente das imagens. Como mencionado, a extração de tamanho dependeria de um controle das condições de aquisição (a câmera deveria estar localizada a uma distância fixa do centro da flor, por exemplo). A localização geográfica não depende de características de cor ou forma e, portanto, está associada a um conceito semântico. Contudo, cabe ressaltar que nem sempre eles serão suficientes para discriminar casos que não foram corretamente classificados com base na análise dos decritores. Veja as informações da Tabela 76 (extraída da Tabela 68).

Note que ambas são da cor amarela, podem estar presentes na Europa e possuem uma variação de tamanho em que um determinado valor pode ser associado a ambas (suponha uma amostra de 20 cm, por exemplo). Este exemplo ilustra a grande complexidade associada aos

processos de classificação, em que erros estão tipicamente presentes. Com a abordagem proposta neste trabalho, pretende-se minimizar sua ocorrência.

Portanto, é possível observar que ambas as abordagens apresentaram problemas: mesmo combinando-as, em alguns casos os problemas relacionados ao *semantic* e ao *sensorial gap* são minimizados mas não eliminados. Uma forma de aperfeiçoar as abordagens se dá pela utilização de um classificador de melhor desempenho, como o SVM (NASCIMENTO et al., 2009) e de uma combinação de descritores mais robustos.

## 5.6 CONSIDERAÇÕES SOBRE O ESTUDO TEÓRICO

A principal vantagem do CBIR em relação a abordagem com ontologias está no tempo de implementação de um sistema do tipo. A restrição de resultados por um limiar possui um tempo de implementação bastante inferior ao da construção de uma ontologia e o algoritmo de busca da mesma, dependendo do domínio do problema.

Uma das desvantagens da recuperação de imagens com ontologia é a dificuldade em especificar e construir uma ontologia. Como analisado em Freddo (2010), fala-se em “Engenharia de Ontologias”, uma tarefa que requer tanto o conhecimento do domínio do problema quanto da terminologia e procedimentos empregados em sua construção. Naquele trabalho discute-se uma metodologia de desenvolvimento denominada *Folkconcept*, cujo objetivo é de auxiliar especialistas de um domínio na construção de uma ontologia de qualidade.

A modelagem com ontologias não elimina completamente problemas relacionados ao *semantic* e *sensorial gap*, já que a etapa de classificação depende apenas do valor dos descritores. Contudo, a modelagem da ontologia torna a comparação de similaridade mais intuitiva ao usuário e relativa ao mundo real. Como pode-se perceber, o atributo “tamanho da imagem” da ontologia leva em conta as dimensões reais do objeto e não da área da região de interesse das imagens, esta última dependendo do foco da lente que realizou a aquisição.

Um dos problemas não solucionados pela abordagem atual e quanto à relevância de cada atributo da ontologia em indicar a similaridade. Não estudou-se favorecer o atributo de cor, por exemplo, em relação aos demais e seu impacto nos resultados. Para o usuário final de um sistema de recuperação de imagens esta funcionalidade poderia ser aproveitada para realizar buscas mais eficientes, restringindo o conjunto de respostas baseado em suas necessidades.

Resumidamente, a tabela 5.6 apresenta um resumo das vantagens e desvantagens de cada técnica.

Técnica	Vantagens	Desvantagens	Propostas de melhoria
Abordagem por CBIR	Implementação mais rápida. Válida para qualquer domínio, embora o desempenho varia para cada descritor	Apresenta <i>semantic</i> e <i>sensorial gap</i>	Redução do <i>semantic</i> e <i>sensorial gap</i>
Abordagem por Ontologias	Combate o <i>semantic</i> e <i>sensorial gap</i>	Dificuldade de implementação, depende do domínio do problema	Melhoria no processo de construção de ontologia

**Tabela 77:** Vantagens de desvantagens das abordagens

## 6 DOCUMENTAÇÃO DE *SOFTWARE*

Como suporte à avaliação de descritores, teste do CBIR e CBIR com ontologia, desenvolveu-se um *software* capaz de automatizar os processos que envolvem essas três etapas. Este capítulo visa apresentá-lo e documentá-lo.

Há apresentação da literatura utilizada no projeto de *software* (6.1), que apresenta as referências utilizadas em seu projeto, como a especificação de requisitos (6.1.1), diagramas UML (6.1.2) e padrões de projeto (6.1.3).

A seção 6.2 apresenta o projeto do *software*, com o levantamento de requisitos (6.2.1), diagramas de caso de uso (6.2.3) e classes (6.2.4).

### 6.1 LITERATURA UTILIZADA NO PROJETO DE *SOFTWARE*

O projeto de *software* utiliza diversos conceitos de engenharia de software, conforme descritos nas seções seguintes.

#### 6.1.1 REQUISITOS

O levantamento de requisitos deste trabalho leva em conta (BASTOS, 2011), diferenciando-os em funcionais e não funcionais. Os requisitos funcionais são relativos ao funcionamento do *software* enquanto os não funcionais relacionam-se aos demais requisitos.

O levantamento de requisitos é apresentado na seção 6.2.1, separados em funcionais 6.2.1.1 e 6.2.2.

#### 6.1.2 UML

A UML (OMG, 2013) é uma linguagem de modelagem utilizada para especificar, visualizar e documentar modelos de *software*. Foi criada por Grady Booch, Ivar Jacobson e Jim Rumbaugh na *Rational Software*.

A UML define um conjunto de diagramas utilizados para auxiliar as tarefas de desenvolvimento de *software*. Seus diagramas são divididos em três conjuntos: Estruturais, Comportamentais e de Interação.

Neste trabalho serão utilizados os diagramas de Classes, Casos de Uso e Sequência, usados para a realização do projeto do *software*.

### 6.1.3 PADRÕES DE PROJETO

Os padrões de projeto são soluções reutilizáveis para problemas comuns de projeto de *software*. Foram propostos pela chamada GoF, composta por Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides.

Os padrões de projeto estão listados em (GAMMA et al., 1995). São ao todo 18 padrões separados em três categorias: Criacionais, Estruturais e Comportamentais. As subseções abaixo listam alguns padrões utilizados neste trabalho.

#### 6.1.3.1 Padrão *Template*

O padrão de projeto *template* é um padrão comportamental. Ele é utilizado para solucionar problemas onde há a necessidade de implementar o esqueleto de um algoritmo, delegando seus passos para subclasses. Seu objetivo é permitir alterar trechos do algoritmo sem alterar sua estrutura.

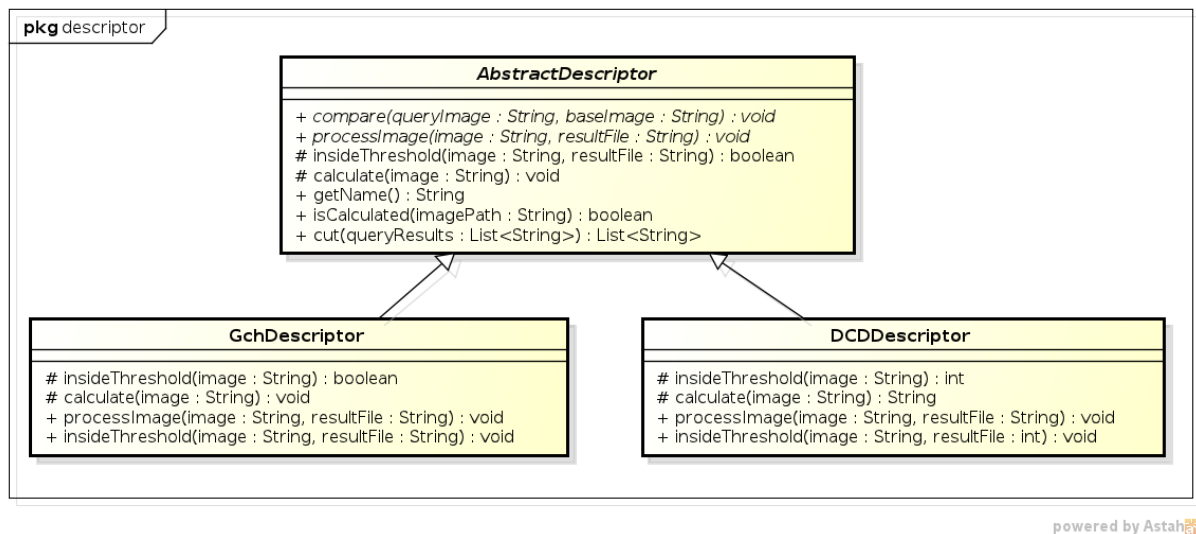
O padrão *template* foi utilizado neste trabalho para a implementação do descritor de imagens. Uma vez que um estudo comparativo pode ter um ou mais descritores é possível definir o esqueleto de um descritor de imagens.

Os métodos *template* são: *compare(queryImage : String, baseImage : String)* e *cut(queryResults : List<String>)* e executam a função de distância e cortam os resultados cuja distância não atenda uma parâmetro. Estes métodos invocam os métodos abstratos *calculate(image: String)* e *insideThreshold(image: String, resultFile : String)*, que serão implementados nas classes derivadas.

#### 6.1.3.2 Padrão DAO

O padrão DAO é um padrão de projeto utilizado em aplicações onde há acesso a um banco de dados ou mecanismo de persistência. Tem como objetivo definir uma interface de chamadas aos dados de forma a isolar a camada de modelo da camada de acesso à aqueles (TACLA, 2011).





**Figura 44: Padrão template**

Pode-se mudar o mecanismo de persistência sem a necessidade de alterações na camada de modelo através do padrão DAO.

Neste trabalho, adotou-se o padrão DAO para gerenciar a entrada e saída de dados de um experimento. A classe *ExperimentDAO* (figura 45) define o conjunto de operações para gerar, carregar e salvar um experimento, através dos métodos *generateExperiment()*, *loadExperiment()* e *saveExperiment()*.

Duas classes concretas herdam métodos de *ExperimentDAO*. São as classes *ExperimentDAOXML* e *ExperimentDAOTable*, a primeira delas gera o arquivo de especificação de experimento e carrega-o, mas não possui a opção de salvar os dados no padrão, já que não há polimorfismo. A classe *ExperimentDAOTable* implementa apenas o método *saveExperiment()*, para salvar os dados de experimento em tabelas no padrão  $\text{L}^{\text{T}}\text{E}^{\text{X}}$ , como discutido no capítulo 4.

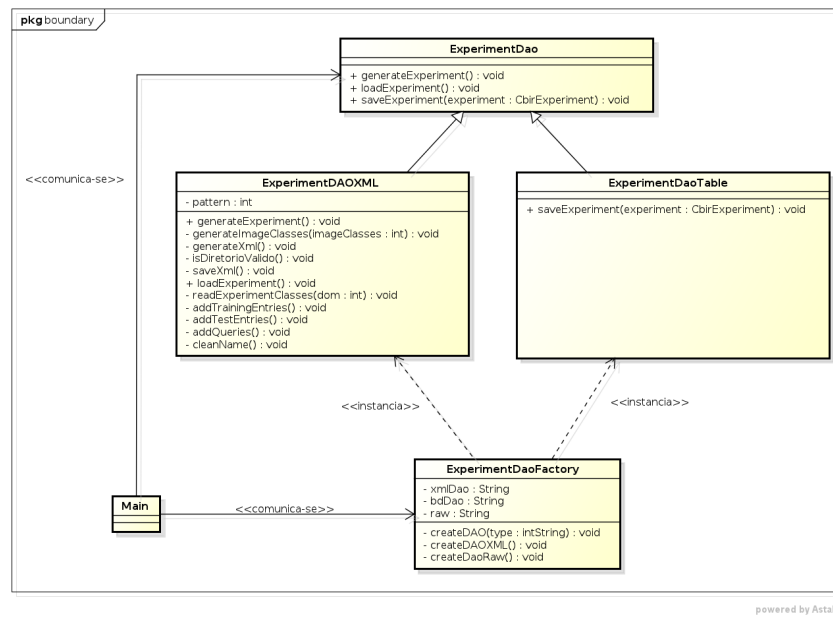
O padrão DAO garantiu ao *software* uma importante característica: A de tornar possível a personalização do formato de saída dos dados de experimento.

## 6.2 PROJETO DE SOFTWARE

Nesta seção é apresentado o projeto de *software*.

### 6.2.1 LEVANTAMENTO DE REQUISITOS

Para o sistema proposto é possível identificar os seguintes requisitos, os quais são classificados em funcionais e não funcionais, conforme (BASTOS, 2011).



**Figura 45: Apresentação do padrão DAO para geração, leitura e escrita de experimentos**

### 6.2.1.1 Requisitos funcionais

Para a construção do software pode-se identificar um ator, o “usuário”. Ele é aquele que deseja avaliar a qualidade de um conjunto de buscas para um dado conjunto de descritores ou de geradores de atributos, no caso do método de aprendizado simbólico. Os requisitos funcionais são:

Para facilitar a separação dos requisitos, iremos separá-los em requisitos funcionais de CBIR e de busca por ontologia. Para a CBIR os requisitos são:

- O *software* deverá permitir ao usuário criar um modelo de experimento, que para uma dada base de imagens separadas por classes distintas, selecionará um conjunto de treinamento e testes (RF-01).
- O *software* deverá carregar um modelo de experimento CBIR salvo, e executar todas as *queries* para um dado conjunto de testes (RF-02).
- O *software* deverá exibir os resultados do experimento CBIR contendo precisão e revocação, separados por classe do objeto (RF-03).
- O *software* deverá exibir a precisão e revocação geral do experimento CBIR (RF-04).
- O *software* deverá exibir para uma dada *query* todas as imagens classificadas como mais similares, separando-as em suas respectivas classes (RF-05).

- O *software* deverá realizar a segmentação de uma base de imagens que contenha a separação da região de interesse (O chamado *ground truth*) (RF-06).

Para a busca de imagens por ontologia, os requisitos funcionais são semelhantes ao CBIR. Para facilitar a execução e a comparação de técnicas, os dados de experimento utilizados serão compatíveis entre os dois métodos. Os itens abaixo listam-nos.

- O *software* deverá carregar um experimento salvo, e executar todas as *queries* na ontologia para um dado conjunto de testes (RF-07).
- O *software* deverá exibir os resultados do experimento de ontologias contendo precisão e revocação, separados por classe do objeto (RF-08).
- O *software* deverá exibir a precisão e revocação geral do experimento de ontologias (RF-09).

### 6.2.2 REQUISITOS NÃO FUNCIONAIS

Os requisitos funcionais do *software* são:

- Os dados do modelo de experimento deverão ser persistidos em um arquivo *xml*. (RNF-01)
- A base de imagens deverá ser separada em uma estrutura de diretórios, sendo cada diretório relativo à uma classe de objetos da mesma base (RNF-02).
- Os resultados de precisão e revocação deverão ser apresentados em arquivos separados e salvos em formato *png* (RNF-03).

### 6.2.3 CASOS DE USO

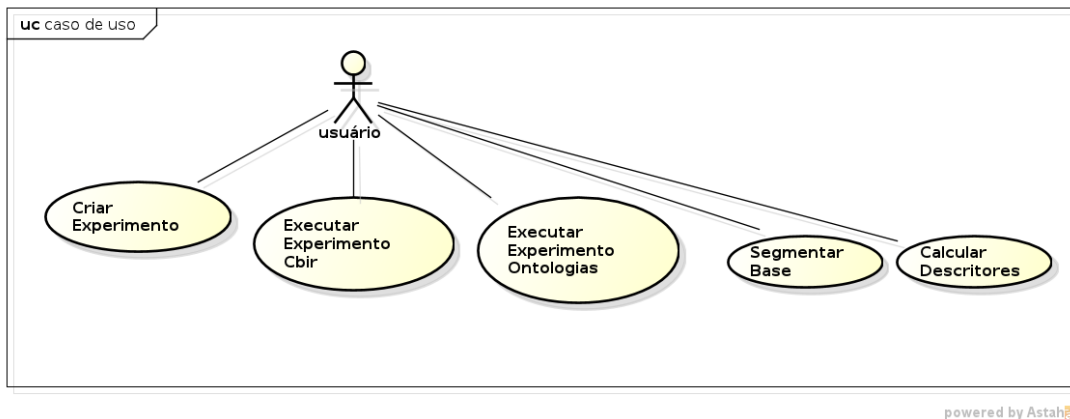
A partir dos requisitos listados acima, são identificados os seguintes casos de uso:

- Criar modelo de experimento, feito por usuário (UC-01).
- Executar experimento CBIR, feito por usuário (UC-02).
- Executar experimento de Ontologias, feito por usuário (UC-03).
- Segmentar base, feito por usuário (UC-04).

- Calcular descritores, feito por usuário (UC-05).

O ator envolvido é o usuário do sistema, que deseja analisar a técnica de busca e seus resultados.

O diagrama de casos de uso pode ser visualizado na figura 46. Para cada caso de uso, suas especificações são fornecidas nas seções abaixo.



**Figura 46: Diagrama de casos de uso do sistema**

**Fonte: Autoria própria.**

#### 6.2.3.1 Caso de uso: Criar Modelo de Experimento

**Ator principal:** Usuário

**Fluxo básico:**

1. O usuário solicita ao sistema a criação de um modelo de experimento.
2. O sistema pede ao usuário que entre com os seguintes dados:
  - (a) Caminho da base
  - (b) Caminho do arquivo de modelo de experimento.
3. O sistema abre o diretório da base de imagens, seleciona cada classe de objetos, separando aleatoriamente 80% dos dados para testes e 20% para testes.
4. O sistema persiste o arquivo de experimento em disco.

**Fluxo alternativo:**

**2a:** Os dados de Caminho da base ou arquivo estão incorretos.

*2.a.1:* O sistema pede ao usuário para reinsserir as informações.

*2.a.2:* Segue para o passo 3 do fluxo básico.

**2b:** O usuário opta por não inserir nenhum dado de caminho ou para a base ou para o caminho do modelo de experimento.

*2.b.1:* O sistema usa os dados padrão de caminho do sistema (pasta */baseImagens*) e */tmp/experiment.xml*.

*2.b.2:* Segue para o passo 3 do fluxo básico.

**Regras de negócio:**

1. O caminho dos diretórios da base e para salvar o modelo de experimento deve ser válido.
2. A base de imagens deve estar separada em subdiretórios, cada qual contendo uma classe de indivíduos.

## 6.2.3.2 Caso de uso: Executar Experimento CBIR

**Ator principal:** Usuário

**Fluxo básico:**

1. O usuário solicita ao sistema executar um experimento CBIR
2. O sistema pede ao usuário que entre com os seguinte dado:
  - (a) Caminho do arquivo de modelo de experimento.
3. O sistema carrega o modelo de experimento em suas entidades.
4. O sistema carrega a lista de descritores
5. O sistema seleciona as imagens de *query* e aplica o descritor nelas uma a uma, encontrando os resultados.
6. O sistema salva os resultados para cada *query* na memória.
7. O sistema calcula a precisão e revocação para cada classe de entrada e depois para todo o experimento.

8. O sistema salva os dados de precisão e revocação em disco.

**Fluxo alternativo:**

**2a:** Os dados de arquivo de modelo de experimento estão incorretos.

*2.a.1:* O sistema pede ao usuário para reinserir as informações.

*2.a.2:* Segue para o passo 3 do fluxo básico.

**2b:** O usuário opta por não inserir nenhum dado de caminho ou para a base ou para o caminho do modelo de experimento.

*2.b.1:* O sistema usa os dados padrão de caminho do sistema (pasta */tmp/experiment.xml*).

*2.b.2:* Segue para o passo 3 do fluxo básico.

**Regras de negócio:**

1. Os gráficos de precisão e revocação serão salvos separadamente por classe de imagens e para o experimento.

6.2.3.3 Caso de uso: Executar Experimento Ontologias

**Ator principal:** Usuário

**Fluxo básico:**

1. O usuário solicita ao sistema executar um experimento Ontologia

2. O sistema pede ao usuário que entre com os seguinte dado:

(a) Se a busca por “tamanho real” da classe será do tipo “igual”, semelhante ou ignorada

(b) Se a busca por “cor” da classe será do tipo “igual”, semelhante ou ignorada

(c) Se a busca por “localização” da classe será do tipo “igual”, semelhante ou ignorada

(d) Se as classes resultantes de cada busca por parâmetro individual serão agrupados via união ou intersecção.

3. O sistema carrega o modelo de experimento em suas entidades.

4. O sistema seleciona as imagens de *query* e aplica os descritores nelas uma a uma, salvando-os em um arquivo de atributos.

5. O sistema obtém a classe para cada imagem de *query*
6. O sistema realiza a busca na ontologia por classes similares
7. Para cada *query* e sua classe encontrada, o sistema procura na ontologia quais classes são mais similares à classe do objeto agrupando-as da forma inserida pelo usuário.
8. O sistema retorna para cada *query* um diretório contendo a própria classe e amostras de imagem resultantes da busca na ontologia.

#### 6.2.3.4 Caso de uso: Segmentar Base

**Ator principal:** Usuário

**Fluxo básico:**

1. O usuário solicita ao sistema segmentar base.
2. O sistema pede ao usuário que entre com os seguintes dados:
  - (a) Caminho da base
  - (b) Caminho dos arquivos de *ground truth*.
3. O sistema abre o diretório da base de imagens e dos arquivos de segmentação.
4. O sistema seleciona uma imagem e seu respectivo *ground truth*, salvando-a na base.

**Fluxo alternativo:**

**2a:** Os dados de Caminho da base ou arquivo estão incorretos.

*2.a.1:* O sistema pede ao usuário para reinserir as informações.

*2.a.2:* Segue para o passo 3 do fluxo básico.

**2b:** O usuário opta por não inserir nenhum dado de caminho ou para a base ou para o caminho do modelo de experimento.

*2.b.1:* O sistema usa os dados padrão de caminho do sistema (pasta */baseImagens*) e */baseImagens/trimaps*.

*2.b.2:* Segue para o passo 3 do fluxo básico.

**Regras de negócio:**

1. O caminho dos diretórios da base e dos arquivos de *ground truth* deve ser válido.

### 6.2.3.5 Caso de uso: Calcular descritores

**Ator principal:** Usuário

**Fluxo básico:**

1. O usuário solicita ao sistema calcular descritores.
2. O sistema pede ao usuário que entre com os seguintes dados:
  - (a) Caminho da base
  - (b) Caminho para salvar a base de descritores.
3. O sistema abre o diretório da base de imagens e dos arquivos de segmentação.
4. O sistema calcula os descritores implementados para a base e salva-os no caminho fornecido para a base de descritores.

**Fluxo alternativo:**

**2a:** Os dados de Caminho da base ou da base de descritores estão incorretos.

*2.a.1:* O sistema pede ao usuário para reinserir as informações.

*2.a.2:* Segue para o passo 3 do fluxo básico.

**2b:** O usuário opta por não inserir nenhum dado de caminho ou para a base ou para o caminho do modelo de experimento.

*2.b.1:* O sistema usa os dados padrão de caminho do sistema (pasta */baseImagens*) e */desc*.

*2.b.2:* Segue para o passo 3 do fluxo básico.

**Regras de negócio:**

1. O caminho dos diretórios da base e da base de descritores deve ser válido.

### 6.2.4 DIAGRAMAS DE CLASSES

Os diagramas de classe da Figura 47 e Figura 48 ilustram a composição das classes que servem de modelo para os descritores. É possível notar que em ambas há uma classe abstrata principal com prefixo *Abstract* que deve ser especializada para a construção de cada descritor individual.

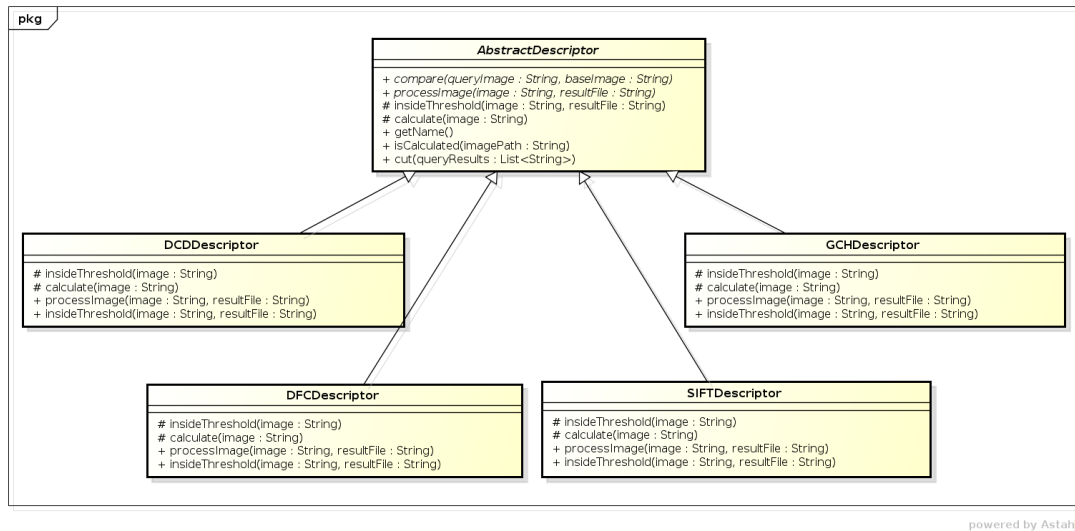


As funções generalizadas para os descritores do CBIR convencional são diferentes das apresentadas nos descritores da ontologia, pois se tratam de parâmetros de busca diferentes para a composição de cada descritor.

No caso do CBIR convencional há a busca de dados numéricos e para o caso da ontologia há a busca por palavras definidoras de características. Abaixo há a descrição das classes para os dois diagramas.

- CbirexperimentHandler é a classe base para o experimento, ela contém referência para os descritores que serão utilizados, a imagem de query e para a configuração do experimento.
- QueryHandler é a classe que encapsula a imagem de query a ser testada no experimento e compartilha com a classe CbirExperimentHandler os dados dos descritores. Adicionalmente contém referência para as imagens de treinamento e teste.
- ImageClass é a classe que encapsula a imagem de query
- AbstractDescriptor é a classe que encapsula do descritor
- CbirExperiment é a classe que encapsula os atributos do experimento
- OntologyExperimentHandler é a classe base para o experimento, ela contém referência para o experimento, para o localizador de termos da ontologia e para a imagem de query.
- OntoQueryHandler é a classe que encapsula a imagem de query a ser testada no experimento e os descritores a serem utilizados
- ImageClass é a classe que encapsula a imagem de query
- AbstractOntologyAttributeFinder é a classe que encapsula os descritores
- ArffGenerator é a classe que encapsula a montagem do arquivo para configuração da ontologia
- Ontoquery é a classe que contém os parâmetros do experimento

O experimento é modelado conforme as classes exibidas na Figura 49 e Figura 50. Os métodos encapsulados nessas classes permitem a execução de um experimento junto à utilização de descritores implementados previamente.



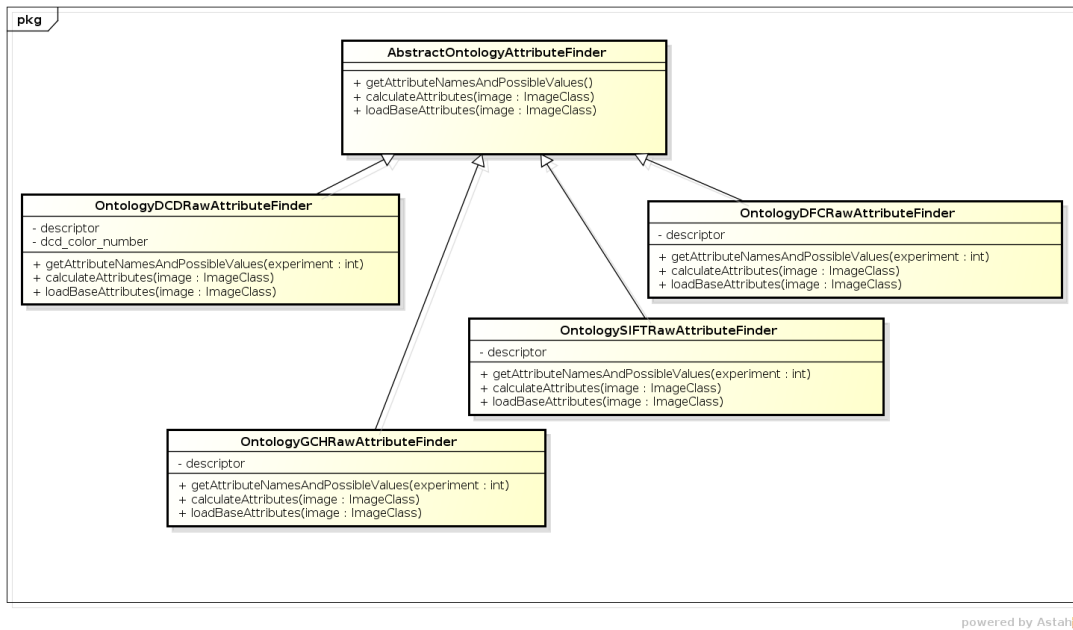
**Figura 47: Classes de descritores**

Nota-se que há semelhanças entre as duas composições de classes dos diagramas e são elas a existência de uma classe de sufixo Handler que manipula o experimento e a dependência das classes ImageClass que encapsula a imagem em si e da classe abstrata que origina os descritores.

Os componentes extras são particulares de cada modelo, como na ontologia são as classes que criam os arquivos que modelam a ontologia e no CBIR a classe que contém as informações do experimento executado.

Abaixo há a descrição das classes para os descritores.

- AbstractDescriptor é a classe base que modela o descritor
- DCDDescriptor é a classe que implementa o descritor DCD
- GCHDescriptor é a classe que implementa o descritor GCH
- DFCDescriptor é a classe que implementa o descritor DFC
- SIFTDescriptor é a classe que implementa o descritor SIFT
- AbstractOntologyAttributeFinder é a classe base dos descritores de ontologia
- OntologyDCDRawAttributeFinder é a classe que implementa o descritor DCD
- OntologyGCHRawAttributeFinder é a classe que implementa o descritor GCH
- OntologySIFTRawAttributeFinder é a classe que implementa o descritor SIFT
- OntologyDFCRawAttributeFinder é a classe que implementa o descritor DFC



**Figura 48: Classes de descritores da ontologia**

### 6.2.5 EXEMPLO DE SAÍDA DE EXECUÇÃO CBIR

O software gera um conjunto de arquivos de suporte à visualização dos parâmetros do experimento. É possível visualizar na Figura 51 a distribuição de pastas que representam cada classe da base de imagens.

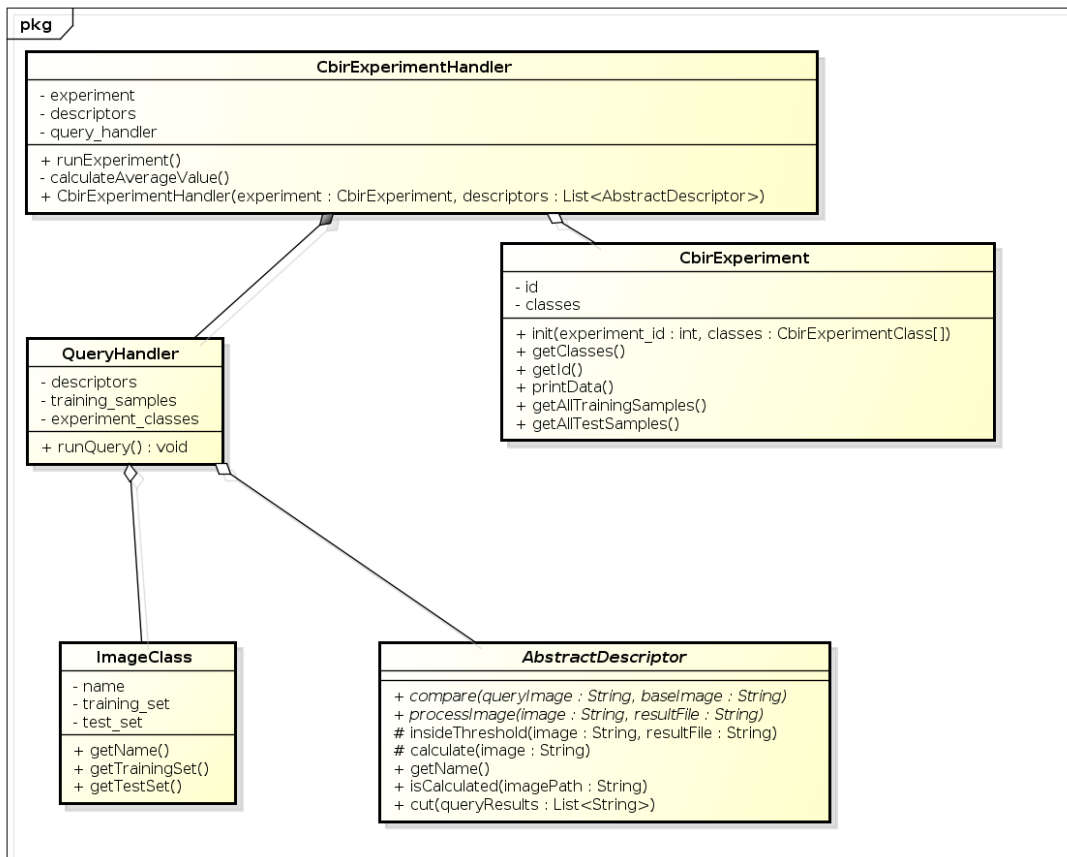
Para cada pasta gerada, conforme Figura 52, há as classes que foram corretamente classificadas e as que tiveram erro de classificação.

Na Figura 53 há um exemplo de arquivo de saída para o experimento. Esse arquivo é gerado automaticamente e contém os parâmetros da execução do CBIR: precisão e revocação.

### 6.3 CONSIDERAÇÕES SOBRE O CAPÍTULO

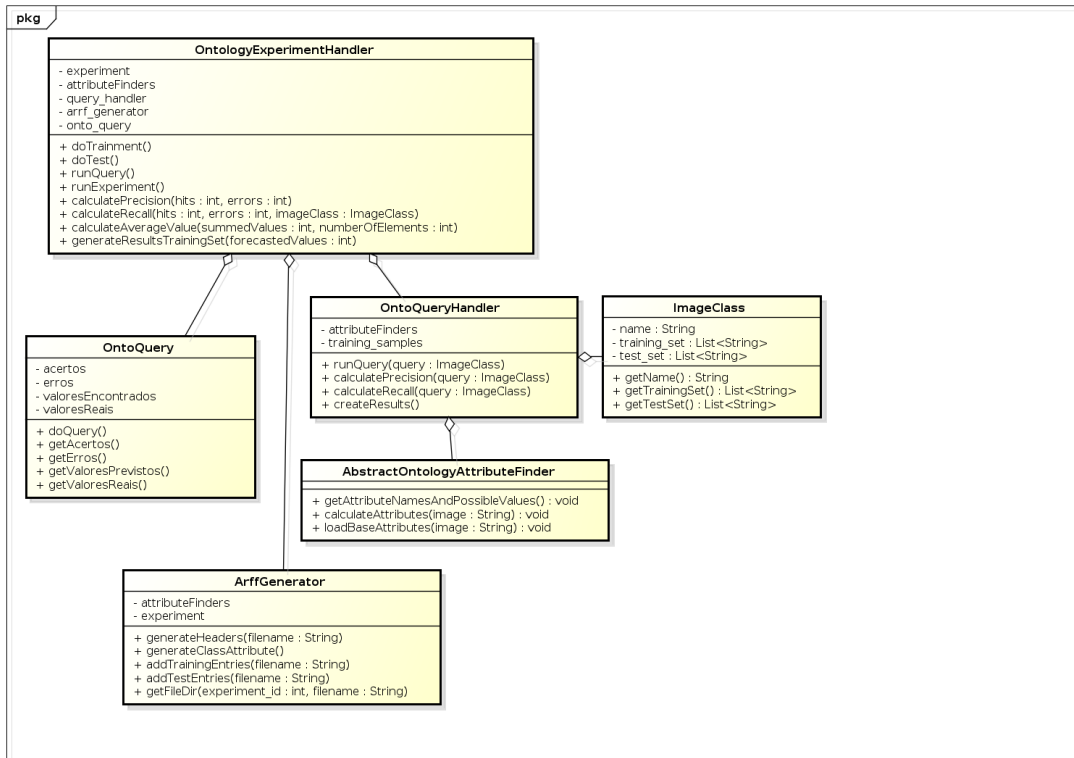
No capítulo 4 discutiu-se o *software* a partir de uma perspectiva mais voltada ao usuário dos experimentos. Este capítulo teve como objetivo apresentar seu projeto e especificação.

A documentação através do levantamento de requisitos, casos de uso e diagramas de classe fornece uma visão mais detalhada dos componentes do *software* para eventuais modificações futuras.



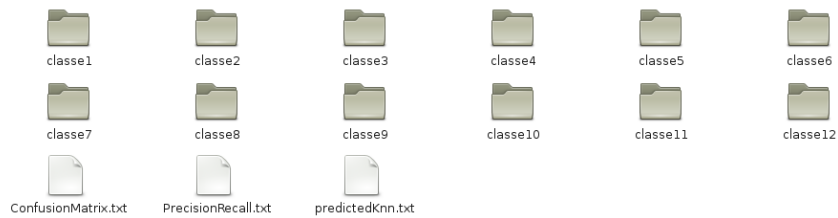
powered by Astah

Figura 49: Classes do experimento CBIR



powered by Astah

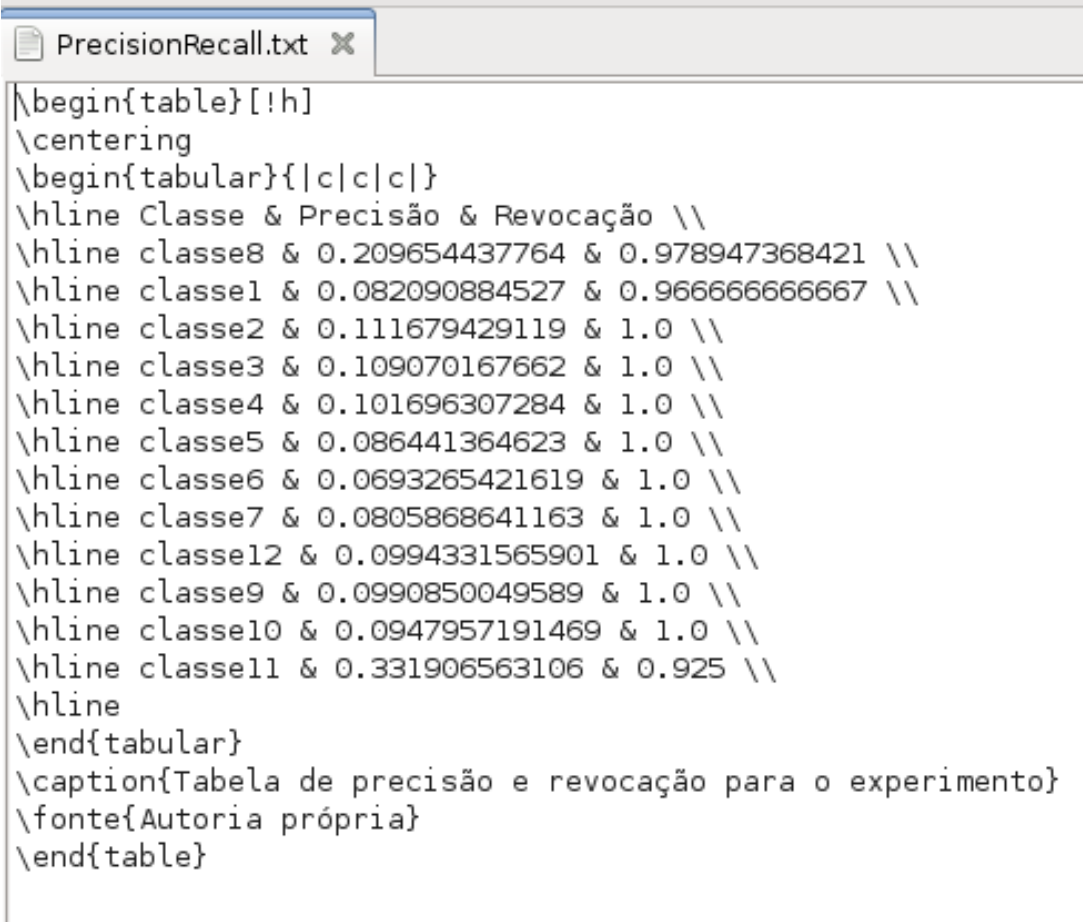
**Figura 50: Classes do experimento CBIR com ontologia**



**Figura 51: Saída do experimento CBIR**



**Figura 52: Acertos e erros por classe**



```

\begin{table}[!h]
\centering
\begin{tabular}{|c|c|c|}
\hline Classe & Precisão & Revocação \\
\hline classe8 & 0.209654437764 & 0.978947368421 \\
\hline classe1 & 0.082090884527 & 0.966666666667 \\
\hline classe2 & 0.111679429119 & 1.0 \\
\hline classe3 & 0.109070167662 & 1.0 \\
\hline classe4 & 0.101696307284 & 1.0 \\
\hline classe5 & 0.086441364623 & 1.0 \\
\hline classe6 & 0.0693265421619 & 1.0 \\
\hline classe7 & 0.0805868641163 & 1.0 \\
\hline classe12 & 0.0994331565901 & 1.0 \\
\hline classe9 & 0.0990850049589 & 1.0 \\
\hline classe10 & 0.0947957191469 & 1.0 \\
\hline classe11 & 0.331906563106 & 0.925 \\
\hline
\end{tabular}
\caption{Tabela de precisão e revocação para o experimento}
\fonte{Autoria própria}
\end{table}

```

**Figura 53:** Arquivo de precisão vs recall gerado automaticamente

## 7 GESTÃO DO PROJETO

### 7.1 GESTÃO DE RECURSOS

Os recursos utilizados no projeto são em grande parte de software, não obstante o hardware se mostra como arcabouço para o desenvolvimento do estudo. Os estudos teóricos, neste projeto estudados, materializados em softwares de simulação de desempenho, contaram com um computador capaz de processar algoritmos voltamos a processamento de imagem.

#### 7.1.1 HARDWARE

O computador utilizado para fornecer condições de processamento de imagens conta com processador Intel Pentium D E5500 com processamento de 3Ghz, HD com 320 GB de espaço e 2 GB de Memória RAM. O hardware acaba sendo necessário para a simulação lógica e deve suportar rotinas de processamento de imagem.

#### 7.1.2 SOFTWARE

Os softwares abrangem todos os recursos lógicos utilizados para materializar os estudos teóricos e envolvem ferramentas baseadas em software livre as quais estão listadas abaixo:

- Sistema Operacional Debian 6.0
- Python 2.7
- Java SDK 1.6
- Python Matplot 1.2.0
- Python Numpy 1.7.0
- Python Scipy 0.11.0
- Python Approximate Nearest Neighbor(ANN) 1.1.2

- Eclipse Juno

Com essas ferramentas, foi possível criar o framework para realizar os experimentos em processamento de imagem através de recuperação por conteúdo e visualização dos resultados em imagens ou tabelas de modo a tornar possível comparações entre implementações e descritores.

### 7.1.3 CRONOGRAMA FINAL

O cronograma final contempla os alinhamentos de escopo, custo e tempo de modo que esses três pilares fundamentem a qualidade dos resultados finais obtidos dos experimentos e objetivos propostos. A figura 54 ilustra o cronograma no qual o projeto está baseado. Na figura 55 está a representação em Gantt do cronograma.

O tempo estimado foi de 1446 horas de trabalho e o tempo realizado foi inferior ao previsto.

	📌	Name	Work	Membro da equipe	Predecessors
1		📁 Mapeamento Ontologia-CBIR	186 hours	Lucas e Solano	
2	📁	Projeto do software de CBIR e Ontologias	72 hours	Lucas	
3		Estrutura dos experimentos	42 hours	Lucas	2
4		Documentação	72 hours	Solano	3
5		📁 Estudo dos descritores	108 hours	Lucas e Solano	
6		Levantamentos dos descritores	72 hours	Lucas	4
7		Integração dos descritores com ontologia	36 hours	Solano	6
8		📁 Implementar algoritmo de query na ontologia	36 hours	Lucas	
9		Implementar e testar algoritmo geral	36 hours	Lucas	7
10		📁 Teste de descritores e execução do experimento	180 hours	Solano	
11		Teste dos descritores DCD, GCH, DFC e SIFT	72 hours	Solano	7
12		Teste do mapeamento das ontologias com descritores	36 hours	Solano	11
13		Experimento para bases separas e análise dos descritores	36 hours	Solano	12
14		Experimento para base completa para descritores DCD e SIFT	36 hours	Solano	13
15		📁 Monografia	444 hours	Lucas	
16		Descrever conceitos teóricos de CBIR	36 hours	Lucas	9
17		Descrever conceitos teóricos de Ontologias	36 hours	Lucas	16
18		Falar de sensorial e semantic gap	36 hours	Lucas	17
19		Capítulo de projeto de software UML e requisitos empregados no desenvolvimento	24 hours	Lucas	18
20		Projetos similares	24 hours	Lucas	19
21		Capítulo de gerência de projetos	72 hours	Lucas	20
22		Estudo comparativo	48 hours	Lucas e Solano	21
23		Revisão da monografia e diagramação de seções	72 hours	Lucas e Solano	33
24		Escrita de considerações por capítulo	48 hours	Lucas e Solano	23
25		Escrita da conclusão final e construção de Slides para apresentação	48 hours	Lucas e Solano	23
26		📁 Melhorias e testes, estudo comparativo prático	528 hours	Solano	
27		Teste dos descritores	72 hours	Solano	14
28		Teste de ontologia	72 hours	Solano	27
29		Teste de software	72 hours	Solano	28
30		Testes com a comparação na ontologia	72 hours	Solano	29
31		Testes com outras classes de imagem da base de imagens	72 hours	Solano	30
32		Descrição dos testes para monografia	72 hours	Solano	31
33		Revisões dos métodos e melhorias	96 hours	Lucas e Solano	32

**Figura 54: Cronograma - Tarefas**

As tarefas foram divididas em seis seções macros que abrangem todas as frentes do projeto, sendo elas Mapeamento Ontologia-CBIR, Estudo dos descritores, Implementar algoritmo de



query na ontologia, Teste de descritores e execução do experimento, Monografia e Melhorias e testes.

Mapeamento Ontologia-CBIR trata de tarefas de criação de software para suporte aos experimentos e integração do algoritmo de ontologia ao framework.

O estudo dos descritores envolve o levantamento da literatura sobre os algoritmos utilizados para extração de características de imagens - descritores.

A fase de implementação de algoritmos de query na ontologia é responsável por fazer a integração entre as imagens de consulta e os termos da ontologia.

Em teste de descritores e execução do experimento são realizados diversos testes para se avaliar os descritores e avaliar o experimento quanto às saídas e comportamento do software.

A monografia envolve as tarefas de escrita e documentação do projeto e melhorias e testes está preocupada com correções pontuais e melhorias encontradas que podem aumentar a eficiência da CBIR.

Melhorias e testes é responsável por implementar novas funcionalidades que possam agregar mais informação ao software e testes para confirmar a melhoria.

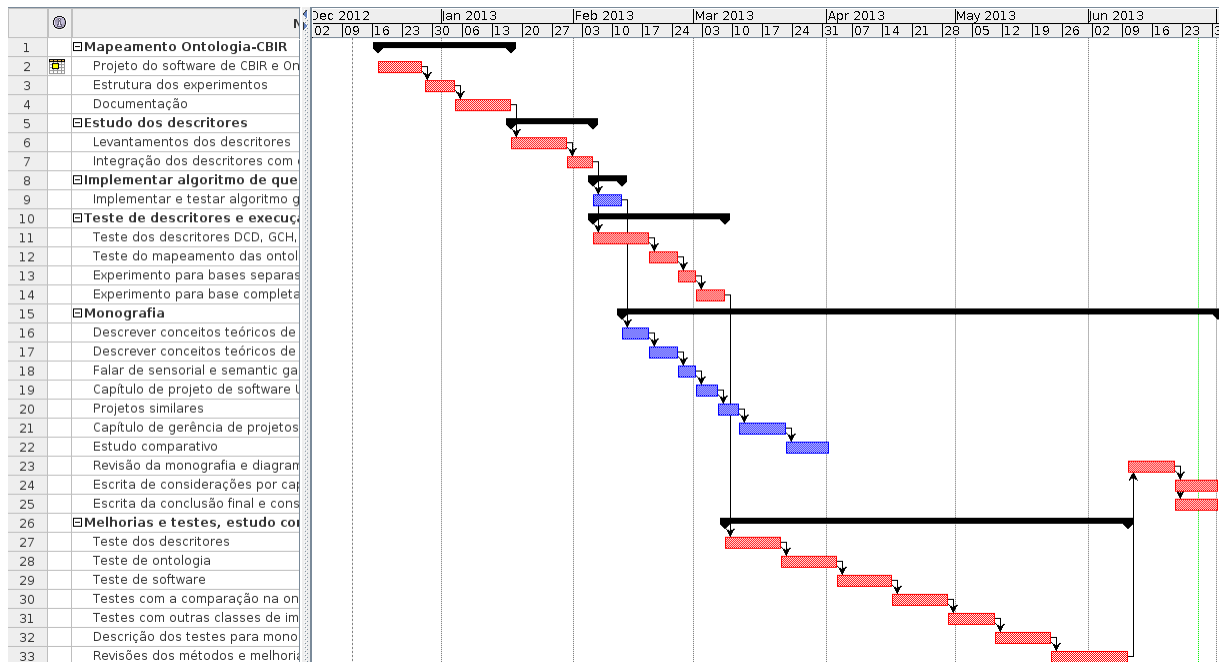


Figura 55: Cronograma - Gantt

#### 7.1.4 CONSIDERAÇÕES

As tarefas foram dimensionadas considerando tempo além do necessário, pois o calendário de trabalho ficou prejudicado devido ao período de greve, recessos e eventos de fim de ano. Assim, houve adição de tempo para cada trabalho levando em consideração períodos de ausência.

A tecnologia Python proporcionou um ganho de produtividade devido a facilidade em se encontrar bibliotecas de processamento de imagem e algoritmos prontos para algumas tarefas que no caso foi o algoritmo de KNN.

O sistema operacional e ambiente para programação não tiveram custo, portanto não houve problemas com licenças e nem gastos.

O conhecimento de linguagens de programação e seus fundamentos, independente de tecnologias ou técnicas particulares permitiu o uso de software livre disponível para o projeto, propiciando ganho de produtividade a custo zero. Este conhecimento facilitou a integração entre as diversas ferramentas como Eclipse, Python e SVN, que auxiliaram todo o projeto a atingir os objetivos propostos.

## 8 CONCLUSÃO

Com o aumento da base de imagens na rede, ocasionado pela popularização de câmeras digitais e dispositivos de captura de imagem, foram necessários meios de se obter e classificar essas imagens com base em critérios estabelecidos (YANG, 2004).

Da necessidade de sistemas mais eficientes para recuperação de imagens, buscou-se nesse trabalho, dentro das técnicas existentes atualmente, propor uma metodologia para melhorar o desempenho da abordagem de recuperação baseada em descritores de cor e forma com a utilização de informações meta-textuais extraídas de uma ontologia, com o objetivo de verificar melhores resultados nesta última.

Com os descritores de cor e forma escolhidos, pode-se montar uma representação numérica das imagens para ser usada na comparação com outras imagens. A base de imagens utilizada propiciou grande diferenciação entre os descritores devido as características variadas de cor e forma presentes nas classes de flores da base.

A proposta de utilização de uma ontologia, pode-se verificar que o *semantic e sensorial gap*, causados pela abordagem CBIR convencional, podem ser reduzidos. Seu principal problema, entretanto, está em sua construção. Para problemas de busca mais abrangentes (exemplo: mecanismo de busca do google), que trabalham com um número elevado de imagens e objetos, torna-se difícil a tarefa de construir uma ontologia abrangente o bastante para a execução da metodologia proposta neste trabalho.

Para os experimentos de CBIR, suas falhas estão associadas às características inerentes do método de extração dos descritores, pois no mapeamento das imagens em números - descritores, perdeu-se a informação semântica e sensorial da região de interesse das imagens. O *semantic e sensorial gap* foram claramente identificados nos experimentos.

Para o suporte ao projeto foram utilizadas ferramentas de *software* livre e construído um sistema capaz de realizar experimentos e avaliar os descritores para a abordagem convencional e por ontologia. Esse sistema propiciou o levantamento facilitado dos resultados e a avaliação do desempenho de cada abordagem, podendo então se obter a conclusão de qual das duas teve

melhor desempenho.

A apresentação do *software* de uma perspectiva de projeto permite a adaptação do mesmo para a realização de outros experimentos envolvendo CBIR. A utilização de padrões de projeto e modelagem UML facilitam modificações futuras.

Finalmente, este trabalho é multidisciplinar, empregando grande parte dos conhecimentos e experiência aprendidos ao longo do curso de graduação. Ele concretiza mais uma vez um de seus propósitos: Utilizar técnicas de computação para solucionar problemas do mundo real.

As seções abaixo apresentam as contribuições do trabalho e trabalhos futuros propostos.

## 8.1 CONTRIBUIÇÕES

Dentre as principais contribuições deste trabalho, estão:

- Análise das características complementares que podem ser abordadas com as duas abordagens de recuperação de imagens - baseada em descritores e em ontologias.
- Documentação do processo de construção de uma ontologia capaz de auxiliar busca de imagens.
- Desenvolvimento de uma ferramenta *software* capaz de auxiliar futuras pesquisas na área de CBIR.

## 8.2 TRABALHOS FUTUROS

Um dos experimentos não explorados no trabalho é a utilização de combinações envolvendo três ou mais descritores, o que por questões de escopo foi descartada, já que o foco deste estudo era o da comparação das técnicas de recuperação de imagens. Um dos trabalhos propostos é a realização de experimentos a partir destas comparações visando perceber se há uma melhora nos resultados.

A utilização de um classificador de melhor desempenho, como o SVM que possui maior robustez na classificação (NASCIMENTO et al., 2009), também pode ser considerada. Um classificador mais sofisticado tenderia a melhorar os resultados para a abordagem baseada em ontologias, permitindo maior precisão na etapa de classificação.

Outro trabalho proposto é o estudo do processo de construção de ontologias aplicada a problemas de reconhecimento de imagens, buscando otimizá-lo. Uma das formas seria uma

análise do método *Folkconcept* (FREDDO, 2010) aplicado em problemas do tipo.

Finalmente, uma vez que um método de construção adequado e um bom classificador fosse utilizado, outro estudo seria combinar bases compostas de vários objetos distintos, utilizando a ontologia na recuperação dos mais similares.

## REFERÊNCIAS

- BASTOS, L. C. **Análise e Projeto de Sistemas**. 2011. Sítio na web. Disponível em: <<http://laudelinobastos.sites.uol.com.br/analiseEC.htm>>.
- BERG, M. de et al. **Computational Geometry: Algorithms and Applications**. Second. Springer-Verlag, 2000. Disponível em: <<http://www.cs.uu.nl/geobook/>>.
- BEYER, K. et al. When is "nearest neighbor" meaningful? In: **In Int. Conf. on Database Theory**. [S.l.: s.n.], 1999. p. 217–235.
- BITTON, D. **MPEG-7 Color Descriptors**. 2013. Disponível em: <<http://tinyurl.com/q83mrn3>>.
- BUGATTI, P. H.; TRAINA, A. J. M. On the effects of distance functions to improve content-based image retrieval. **CTD - XXII Concurso de Teses e Dissertações**, v. 1, p. 105–112, 2009.
- CORCHO, O. et al. Building legal ontologies with methontology and webode. In: **Law and the Semantic Web, number 3369 in LNAI**. [S.l.]: Springer-Verlag, 2005. p. 142–157.
- DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: **Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01**. Washington, DC, USA: IEEE Computer Society, 2005. (CVPR '05), p. 886–893. ISBN 0-7695-2372-2. Disponível em: <<http://dx.doi.org/10.1109/CVPR.2005.177>>.
- DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: **In ICML '06: Proceedings of the 23rd international conference on Machine learning**. [S.l.]: ACM Press, 2006. p. 233–240.
- FENG, D.; SIU, W.; ZHANG, H. **Multimedia Information Retrieval and Management: Technological Fundamentals and Applications**. Springer, 2003. (Signals and Communication Technology). ISBN 9783540002444. Disponível em: <<http://books.google.com.br/books?id=qn54Qv35sm8C>>.
- FLORIDATA. **Floridata**. 2013. Disponível em: <<http://www.floridata.com/>>.
- FRANCISCANI, J. de F. **Redução do Gap Semântico em CBIR utilizando Ontologia de Objeto**. 01 2013. Disponível em: <<http://www.facom.ufu.br/posgrad/wd2008/artigos/Juliana.pdf>>.
- FREDDO, A. **Folkconcept : método de suporte à modelagem conceitual de ontologias a partir da aquisição de conhecimentos de folksonomias**. 2010. Disponível em: <<http://tinyurl.com/lfdwl6k>>.
- GAMMA, E. et al. **Design patterns: elements of reusable object-oriented software**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN 0-201-63361-2.

GRUBER, T. R. A translation approach to portable ontology specifications. **KNOWLEDGE ACQUISITION**, v. 5, p. 199–220, 1993.

Guizzardi, G. **Ontological foundations for structural conceptual models**. Tese (Doutorado), Enschede, 2005. Disponível em: <<http://doc.utwente.nl/50826/>>.

HYVÖNEN, E.; STYRMAN, A.; SAARELA, S. Ontology-based image retrieval. In: . [S.l.: s.n.], 2002. p. 15–27.

JR, J. A.; TORRES, R.; GOLDENSTEIN, S. Sift applied to cbir. **Revista de Sist. de Inf. da Fac. Salesiana Maria Auxiliadora**, v. 4, p. 41-48, 2009. ISSN 1983-5604.

KATO, T. Database architecture for content-based image retrieval. In: **Image Storage and Retrieval Systems**. San Jose, California: [s.n.], 1992.

LIMA, E. S. de. K-nearest neighbor (knn). **PUCPR ? Inteligência Artificial**, 2008.

LIU, X. Supervised classification and unsupervised classification. **ATS 670 Class Project**, 2005.

MAN, W. K. **Content Based Image Retrieval Using MPEG-7 Dominant Color Descriptor**. 2004.

MARQUES, O.; MAYRON, L. M.; BORBA, G. B. **On the Potential of Incorporating Knowledge of Human Visual Attention into CBIR Systems**. 2006.

MEZARIS, V.; KOMPATSIARIS, I.; STRINTZIS, M. G. Region-based image retrieval using an object ontology and relevance feedback. In: **Eurasip Journal on Applied Signal Processing**. [S.l.: s.n.], 2004. p. 886–901.

NASCIMENTO, R. F. F. et al. **O algoritmo Support Vector Machines (SVM): avaliação da separação ótima de classes em imagens CCD-CBERS-2**. Abril 2009.

NILSBACK, M.-E.; ZISSERMAN, A. Automated flower classification over a large number of classes. In: **Proceedings of Sixth Indian Conference on Computer Vision, Graphics & Image Processing**. [S.l.: s.n.], 2008. p. 722–729.

NILSBACK, M.-E.; ZISSERMAN, A. Delving deeper into the whorl of flower segmentation. **Image Vision Comput.**, Butterworth-Heinemann, v. 28, n. 6, p. 1049–1062, 2010. ISSN 0262-8856.

NILSBACK, M.-E.; ZISSERMAN, A. **Flower Datasets**. 10 2012. Disponível em: <<http://www.robots.ox.ac.uk/vgg/data/flowers/>>.

OHM, J. et al. The mpeg-7 colour descriptor. **IEEE Transactions on Circuits and Systems for Video Technology**, 2001.

OMG. **Introduction To OMG's Unified Modeling Language**. 04 2013. Disponível em: <<http://www.omg.org/gettingstarted/>>.

OSHIRO, S. H. **Um Ambiente Inteligente para Recuperação de Imagens**. Dissertação (Mestrado) — Instituto Militar de Engenharia, Rio de Janeiro, 2008.

- PENATTI, O. **Estudo comparativo de descritores para recuperação de imagens por conteúdo na web**. Dissertação (Mestrado) — Universidade Estadual de Campinas. Instituto de computação., Campinas, 2009. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=000444929>>.
- REENSKAUG, T. **Models - Views - Controllers**. [S.l.], 1979. Disponível em: <<http://heim.ifi.uio.no/trygver/themes/mvc/mvc-index.html>>.
- RESEARCH, S. C. for B. I. **The Protegé Ontology Editor and Knowledge Acquisition System**. 2011. Disponível em: <<http://protege.stanford.edu/>>.
- SHARMA, N.; RAWAT, P. **Efficient Cbir Using Color Histogram Processing**. Lambert Academic Publishing, 2012. ISBN 9783659107832. Disponível em: <<http://books.google.com.br/books?id=1ID7ugAACAAJ>>.
- SILVA, G. P. M. P. Mayra Luíza Marques da. Avaliação de algoritmos de classificação supervisionada para imagens do cbers-2 da região do parque estadual do rio doce-mg. **Anais XIII Simpósio Brasileiro de Sensoriamento Remoto**, p. 6223–6228, 2007.
- SINGHA, M.; K.HEMACHANDRAN. Content based image retrieval using color and texture. **Signal & Image Processing**, p. 39–57, 2012.
- SMEULDERS, A. W. M. et al. Content-based image retrieval at the end of the early years. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, p. 1349–1380, 2000.
- SMITH, J. R.; CHANG, S.-F. Visualeek: a fully automated content-based image query system. In: **Proceedings of the fourth ACM international conference on Multimedia**. New York, NY, USA: ACM, 1996. (MULTIMEDIA '96), p. 87–98. ISBN 0-89791-871-1. Disponível em: <<http://doi.acm.org/10.1145/244130.244151>>.
- TACLA, C. A. **Design Patterns Parte 4 : Padrões Factory e DAO**. May 2011.
- TING, K. Confusion matrix. In: SAMMUT, C.; WEBB, G. (Ed.). **Encyclopedia of Machine Learning**. [S.l.]: Springer US, 2010. p. 209–209.
- TORRES, R. D. S.; FALCAO, A. X. Content-based image retrieval: Theory and applications. **Revista de Informática Teórica e Aplicada**, v. 13, p. 161–185, 2005.
- TRAN, L. V. **Efficient Image Retrieval with Statistical Color Descriptors**. 205 p. Tese (Doutorado) — Linköping UniversityLinköping University, Department of Science and Technology, The Institute of Technology, 2003.
- VELTKAMP, R. C.; TANASE, M. **Content-based image retrieval systems: A survey**. [S.l.], 2000.
- W3C. **Ontologies**. 2013. Disponível em: <<http://www.w3.org/standards/semanticweb/ontology>>.
- WANG, S.; WANG, S. **A Robust CBIR Approach Using Local Color Histograms**. [S.l.], 2001.
- WILSON, D. R.; MARTINEZ, T. R. Improved heterogeneous distance functions. In: **Journal of Artificial Intelligence Research** 6. [S.l.: s.n.], 1997. p. 1–34.



YANG, C. C. Content-based image retrieval: A comparison between query by example and image browsing map approaches. **J. Information Science**, p. 254–267, 2004.

ZHANG, D.; LU, G. **Review of shape representation and description techniques**. 2004.