

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE COMPUTAÇÃO DO CURSO DE ANÁLISE E  
DESENVOLVIMENTO DE SISTEMAS

VICTOR MOURA OLIVEIRA

**ESTUDO E ANÁLISE DE MÉTODOS DE REALIMENTAÇÃO DE  
RELEVÂNCIA NO CONTEXTO EDUCACIONAL**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO  
2015

VICTOR MOURA OLIVEIRA

## **ESTUDO E ANÁLISE DE MÉTODOS DE REALIMENTAÇÃO DE RELEVÂNCIA NO CONTEXTO EDUCACIONAL**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho De Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR / Campus Cornélio Procópio, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Pedro Henrique Bugatti

CORNÉLIO PROCÓPIO  
2015

## **AGRADECIMENTOS**

Em primeiro lugar, agradeço ao meu orientador Prof. Dr. Pedro Henrique Bugatti pela oportunidade, sabedoria e ajuda disponibilizada durante toda essa jornada.

Agradeço também aos meus pais pelo apoio incondicional, pois sem eles acredito que seria muito difícil ter chegado aqui.

A coordenação do curso, em especial a Profa. Dra. Érica Ferreira De Souza e a Profa. Ma. Adriane Carla Anastácio da Silva por toda a cooperação.

A todos meus amigos, pelo apoio.

## RESUMO

OLIVEIRA, Victor Moura. **Estudo e análise de métodos de realimentação de relevância no contexto educacional**. 2015. 53 f. Trabalho de Diplomação (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

A recuperação de imagens por conteúdo tornou-se uma importante área nos últimos tempos, sendo alvo de muitas pesquisas e, com isso, a elaboração de novas técnicas e conceitos nesta. Apesar disso, muitos problemas ainda são encontrados no processo de recuperação de imagens. O presente trabalho demonstra um estudo geral sobre a recuperação de imagens por conteúdo e tem como objetivo aplicar métodos de realimentação de relevância em um sistema de contexto educativo, a fim de comparar as diferentes técnicas e verificar qual gera o melhor resultado.

**Palavras-chave:** Recuperação de imagens por conteúdo. Gap semântico. Realimentação de relevância. Sistema de contexto educativo.

## ABSTRACT

OLIVEIRA, Victor Moura. **Study and analysis of methods for relevance feedback in educational context**. 2015. 53 f. Trabalho de Diplomação (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Content-based image retrieval has become an important area in recent years, being subject of much research and, thus, the development of new techniques and concepts. Nevertheless, many problems are still open in the image retrieval process. In the present work it is proposed a general study on content-based image retrieval aiming at applying methods relevance feedback into a system of educational context in order to compare the different techniques and see which generates the best result.

**Keywords:** Content-based image retrieval. Semantic gap. Relevance feedback. System of educational context.

## LISTA DE FIGURAS

Figura 1 - Arquitetura do sistema CBIR [TORRES; FALCÃO, 2006].....	16
Figura 2 - Ilustração do processo de consulta [PENATTI, 2009].....	17
Figura 3 - Histograma de cor de uma textura [CASTAÑON, 2003] .....	20
Figura 4 - Exemplos de texturas com diferentes rotações [TORRES; FALCÃO, 2008] .	20
Figura 5 - Exemplos de formas representadas por folhas de árvores [DA FONTOURA COSTA; CESAR JR, 2000] .....	21
Figura 6 - Arquitetura de um Sistema CBIR com Realimentação de Relevância [DA SILVA, 2011] .....	24
Figura 7 - Técnica de Query Point Movement utilizando a distância Euclidiana [MARQUES, 2006] .....	25
Figura 8 - Exemplo de diferentes rotações possíveis que podem ser implementadas [MARQUES, 2006] .....	27
Figura 9 – Exemplo de imagens para cada uma das classes da base.....	29
Figura 10 - Diagrama Entidade-Relacionamento .....	30
Figura 11 - Modelo Entidade-Relacionamento .....	31
Figura 12 - Diagrama de Classes.....	32
Figura 13 – Tela principal do sistema.....	34
Figura 14 - Tela de seleção da imagem de consulta.....	35
Figura 15 - Tela principal após a seleção de uma imagem .....	36
Figura 16 - Seleção das imagens relevantes/irrelevantes.....	37
Figura 17 - Consulta refinada após a seleção das imagens relevantes/irrelevantes .....	38
Figura 18 – Imagem de um pássaro (1); Imagem de um tigre (2); Imagem de um lagarto (3); Imagem de um jacaré (4); Imagem de um cachorro (5) .....	44
Quadro 1 - Algoritmo de cálculo de distância .....	39
Quadro 2 - Algoritmo de Rocchio .....	41
Gráfico 1 - Gráfico de precisão média para as consultas iniciais dos extratores.....	46
Gráfico 2 - Gráfico de precisão média para as consultas refinadas dos extratores.....	46
Gráfico 3 - Comparativo entre a precisão média das consultas iniciais e refinadas dos extratores .....	47

## LISTA DE TABELAS

Tabela 1 - Testes de precisão sobre a imagem 1 .....	45
Tabela 2 - Testes de precisão sobre a imagem 2 .....	45
Tabela 3 - Testes de precisão sobre a imagem 3 .....	45
Tabela 4 - Testes de precisão sobre a imagem 4 .....	45
Tabela 5 - Testes de precisão sobre a imagem 5 .....	45

## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ACC	<i>Color Correlogram</i>
BIC	<i>Border/interior pixel classification</i>
CBIR	<i>Content-based image retrieval</i>
CSS	<i>Curvature Scale Space</i>
JAC	<i>Joint auto-correlogram</i>
QPM	<i>Query Point Movement</i>
RR	<i>Relevance Feedback</i>
SGDB	<i>Sistema de Gerenciamento de Banco de Dados</i>



## SUMÁRIO

1	INTRODUÇÃO .....	10
1.1	PROBLEMA .....	10
1.2	JUSTIFICATIVA .....	11
1.3	OBJETIVOS .....	12
1.4	ORGANIZAÇÃO DO TEXTO .....	13
2	FUNDAMENTAÇÃO TEÓRICA .....	14
2.1	APRENDIZAGEM DE CONCEITOS POR MEIO DE IMAGENS .....	14
2.2	RECUPERAÇÃO DE IMAGENS POR CONTEÚDO .....	16
2.3	FUNÇÕES DE DISTÂNCIA .....	17
2.4	EXTRAÇÃO DE CARACTERÍSTICAS .....	18
2.4.1	COR .....	19
2.4.2	TEXTURA .....	20
2.4.3	FORMA .....	21
2.5	AVALIAÇÃO DE DESEMPENHO .....	22
2.6	REALIMENTAÇÃO DE RELEVÂNCIA .....	22
2.6.1	MÉTODOS DE REALIMENTAÇÃO DE RELEVÂNCIA .....	24
3	METODOLOGIA .....	28
3.1	BASE DE IMAGENS .....	28
3.2	TECNOLOGIAS .....	29
3.3	MODELAGEM .....	30
3.3.1	DIAGRAMA ENTIDADE-RELACIONAMENTO .....	30
3.3.2	MODELO ENTIDADE-RELACIONAMENTO .....	31
3.3.3	DIAGRAMA DE CLASSES .....	32
3.4	SISTEMA DESENVOLVIDO .....	34
3.5	IMPLEMENTAÇÃO .....	38
3.6	CONSIDERAÇÕES FINAIS .....	42
4	RESULTADOS E DISCUSSÕES .....	44
4.1	TESTES DE PRECISÃO .....	44
4.2	RESULTADOS OBTIDOS .....	45
4.3	CONSIDERAÇÕES FINAIS .....	47
5	CONCLUSÃO .....	49
5.1	TRABALHOS FUTUROS .....	49
	REFERÊNCIAS .....	51

## 1 INTRODUÇÃO

Com o avanço da tecnologia e a consolidação da internet, o acesso, armazenamento e manipulação de grande quantidade de dados se tornou comum e, com isso, vem a necessidade de criar técnicas para recuperar esses dados de forma rápida e eficiente. Dentre todos os tipos de arquivos multimídia – textos, imagens, áudios e vídeos, as imagens são as que mais tem importância, pois não só é o tipo de arquivo mais utilizado depois do texto, como também é o mais utilizado para representar e recuperar outros arquivos multimídia como vídeos [RUI, 1998].

Atualmente, são várias as áreas em que a recuperação de imagens por conteúdo é a principal atividade, onde pode-se citar: gerenciamento de galerias de arte e museus, desenho arquitetural e de engenharia, desenho interior, sistemas de informação geográfica, previsão do tempo, sistemas de armazenamento de imagens e de comunicação em geral, etc. [GUDIVADA; RAGHAVAN, 1995].

Diante disso, o sistema de recuperação de imagens por conteúdo, *Content-based image retrieval* (CBIR), foi criado com o objetivo de englobar técnicas que resolvam o problema de recuperação das imagens em bases de dados.

### 1.1 PROBLEMA

Um dos principais problemas encontrados no CBIR diz respeito ao chamado gap semântico (*semantic gap*) [SMEULDERS, 2000], em que o mesmo ocorre devido à enorme discrepância entre a representação da imagem gerada pela máquina e o conceito de similaridade da visão humana. Tal discrepância ocorre pelo fato das imagens representadas no sistema CBIR serem diferentes da realidade, já que são baseadas nas características de baixo nível da imagem.

Visto que as imagens são analisadas pelo usuário, é possível com que ocorra uma descontinuidade semântica (gap semântico), pois os mesmos podem julgar certas imagens não relevantes como relevantes (ou vice versa), seja pela falta de bons descritores de baixo nível de uma imagem de consulta condizente, ou pela própria análise errada sobre as imagens retornadas, como explicado acima.

Para tentar corrigir esse problema, certas medidas foram criadas como, por exemplo, a classificação e/ou agrupamento das imagens da base de dados baseado em um critério específico (seja uma característica, o tipo do descritor, etc.). Tal medida aprimora a consulta até certo ponto, mas não resolve o problema completamente. Diante disso, dois temas se tornaram importantes fontes de pesquisa em CBIR: a participação do usuário nas estruturas de indexação e na recuperação das imagens da base de dados. Este último se transformou em um processo que é conhecido como realimentação de relevância [TORRES; FALCÃO, 2006].

Inicialmente utilizada para a recuperação de textos, a técnica de realimentação de relevância tem como objetivo permitir que o usuário expresse sua necessidade na consulta de forma mais fácil, sem ser preciso recorrer as especificações de baixo nível para a representação de imagens. Para tanto, a cada busca (iteração) do sistema, a técnica de realimentação de relevância tenta aprender, a partir das imagens indicadas pelo usuário, quais são as características visuais da imagem que mais definem as imagens relevantes. Assim, uma nova consulta é realizada novamente até que, após determinado número de iterações, o sistema começará a retornar as imagens mais relevantes para aquela busca.

## **1.2 JUSTIFICATIVA**

No sistema de recuperação de imagens por conteúdo, cada imagem é descrita (e por consequência, analisada) pela associação de um conjunto de características (informações), tais como: segmentos de borda, regiões, objetos segmentados nas imagens, entre outros. A partir disso, o sistema CBIR faz a comparação entre duas imagens por uma função de distância, que mensura a similaridade entre as características extraídas das imagens. A extração dessas características é realizada pelo descritor.

Como resultado, são recuperadas imagens que o processo de recuperação considera como semelhantes para a consulta, a partir da imagem fornecida pelo usuário e pelos descritores utilizados. Posteriormente, o usuário pode julgar as imagens

retornadas como relevantes para o refinamento da(s) próxima(s) consulta(s), sendo que esse fato caracteriza o processo de realimentação de relevância.

Dito isso, o objetivo dessa proposta é a criação de um sistema no contexto educacional em que será implementado o sistema de recuperação de imagens por conteúdo em conjunto com o processo de realimentação de relevância, a fim de auxiliar os alunos em seu aprendizado pela apresentação de imagens de diferentes conceitos.

A partir do processo de realimentação de relevância e seu módulo de aprendizagem, o aluno poderá selecionar as imagens que ele julga como relevantes e continuar realizando novas consultas até que o resultado trazido esteja condizente com a imagem de consulta, o que indicará durante o processo se o mesmo está escolhendo as imagens corretas.

Assim, o sistema proposto irá oferecer uma maneira interativa e iterativa para o aluno aprender novos conceitos a partir de diferentes imagens de consulta, enquanto também facilitará a vida do professor no processo de ensino.

### **1.3 OBJETIVOS**

O presente trabalho visa o estudo e análise de métodos do processo de realimentação de relevância no contexto educacional. Para tanto, serão elencadas técnicas de realimentação e as mesmas serão calibradas de acordo com o contexto em questão.

Como objetivos específicos, espera-se:

- Análise de diferentes técnicas de realimentação de relevância atreladas ao contexto educacional.
- Geração de um sistema que permita tais análises em ambiente real.
- Validar a eficácia de tal abordagem para o ensino de conceitos por meio de imagens.

## **1.4 ORGANIZAÇÃO DO TEXTO**

No presente capítulo foram elencadas a motivação, justificativa e objetivos relacionados ao trabalho em questão.

O Capítulo 2 apresenta a fundamentação teórica do trabalho, explicando como se dá o processo de aprendizagem de conceitos por meio de imagens pelo aluno, além de conter uma explicação detalhada do processo de recuperação de imagens por conteúdo, bem como do processo de realimentação de relevância.

O Capítulo 3 elucida a metodologia do trabalho, apresentando a base de imagens utilizada e suas tecnologias, bem como demonstra a modelagem do sistema de maneira detalhada, além de descrever o funcionamento do sistema desenvolvido e como o mesmo foi implementado.

Já o Capítulo 4 demonstra os resultados e discussões do trabalho, apresentando os testes de precisão realizados, os resultados obtidos bem como as conclusões sobre os mesmos.

Por fim, o Capítulo 5 apresenta a conclusão final e sugestões futuras para o trabalho desenvolvido.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 APRENDIZAGEM DE CONCEITOS POR MEIO DE IMAGENS

Atualmente, os estudos de comunicação tem-se focado muito no vocabulário, esquecendo-se do mundo visual [SANTOS, 2002]. Apesar dos professores de qualquer área de ensino reconhecerem a importância do ensino pela imagem, o uso da imagem como recurso visual nas salas de aula ainda é muito limitado, onde tem-se como predominância o ensino expositivo por meio de palavras. Isso acontece não pelo simples fato do professor preferir ensinar por palavras, mas sim por o mesmo considerar o ensino por imagens como uma ameaça: hoje em dia, a nova geração de alunos possui uma familiaridade muito grande com imagens, em contraste com o professor. Dessa forma, os professores acabam se sentindo perdidos ao usar a imagem como ensino e, como consequência, acabam utilizando as mesmas como mera ilustração/motivação, ao invés de utilizar realmente como um meio de aprendizagem [LENCASTRE, 2003].

Para o ensino pela imagem, esta deve ser o instrumento de comunicação, informação e de conhecimento [DUBORGEL, 1992]. A imagem é, dessa forma, considerada uma linguagem específica, com valor próprio, o que ainda não é uma realidade apesar de vivermos em uma sociedade dominada pela imagem [LENCASTRE, 2007].

Entretanto, existem algumas regras importantes para o ensino pela imagem, sendo que primeiramente recomenda-se uma exibição dupla da imagem pelo aluno antes do processo de leitura, onde é realizada primeiramente uma visualização da imagem de forma espontânea – chamada de “vagabundagem visual”, sem a mediação do professor [MODERNO, 1992] e, posteriormente, é solicitado que os alunos realizem uma análise mais profunda da imagem com a ajuda do professor, chamado de “inventário da imagem”, onde descreve-se verbalmente a imagem para que assim clarifique-se o conteúdo da imagem ao focalizar nos detalhes que importam [LENCASTRE, 2003].

Assim, o ensino pela imagem é mais do que simplesmente visualizar uma imagem: é um processo de leitura, análise e reflexão sobre a mesma, ou seja, de compreensão. Para este processo, Donis A. Dondis, em seu livro “Sintaxe da linguagem

visual”, elaborou a ideia de “alfabetismo visual”, que é a capacidade dos indivíduos compreenderem uma representação visual [DONDIS, 1997], sendo que o uso desse termo aplica-se as práticas de aprendizagem e técnicas para o entendimento das imagens. Diante disso, é correto afirmar que a linguagem visual, assim como a linguagem verbal, também tem seu código/sistema próprio [CARNEIRO; MANINI, 2009].

Ao alfabetizar visualmente um aluno, estamos ensinando-o a realizar uma leitura das imagens, ou seja, a observar uma imagem em detalhes, com exatidão. Esse processo é realizado através do professor, que deve ensinar o aluno a ver e ensinar a pensar [LENCASTRE, 2007]. Para que isso ocorra, a leitura correta de uma imagem exige dois aspectos: a leitura denotativa e a conotativa. A leitura denotativa é literal, ou seja, refere-se a simples listagem dos elementos (pessoas, objetos, etc.) na imagem, enquanto a leitura conotativa é codificada, exigindo um conhecimento prévio do aluno (saber cultural) para a sua interpretação própria [CARNEIRO; MANINI, 2009]. Assim, pela leitura detalhada sobre a imagem a partir dos aspectos supracitados, o aluno pode compreendê-la por completo.

Ainda sobre o processo de leitura de imagens, podemos classificá-lo em duas teorias: a teoria tipográfica e a teoria de Gestalt. A primeira analisa a imagem similar ao processo de leitura de um texto, enquanto a segunda tem como objetivo destacar, à primeira vista, o tema principal da imagem e posteriormente seus diferentes núcleos de interesses [LENCASTRE, 2007].

As imagens, como as histórias, nos formam e nos informam, além de serem poderosas formas de comunicação. Com base nisso, podemos observar a importância que a imagem tem no processo de ensino, em que quando bem utilizada auxilia o aluno na compreensão dos assuntos abordados em sala de aula, assim como complementa a informação textual presente nessa. Além disso, como atualmente as imagens estão presentes na maioria dos meios de comunicação que cercam os alunos, seu uso em sala de aula tornou-se indispensável.

Entretanto, para que a imagem sirva como forma de ensino, é necessário com que o professor reconheça a importância desta para o aprendizado, utilizando-a de forma correta em sala de aula, já que muitos a utilizam apenas como mera ilustração/motivação.

## 2.2 RECUPERAÇÃO DE IMAGENS POR CONTEÚDO

O sistema de recuperação de imagens por conteúdo é uma técnica focada na similaridade das imagens. Normalmente utilizada em grandes bases de dados de imagens, uma busca em CBIR é realizada por meio de um padrão de consulta informado pelo usuário (geralmente uma imagem), enquanto que a comparação das imagens em si é feita através do uso de descritores [TORRES; FALCÃO, 2006].

Como pode ser visto na Figura 1, a arquitetura CBIR consiste em dois processos principais: a inserção de dados e o processamento das consultas. Na inserção de dados, o objetivo é extrair informações de cada imagem, gerando e organizando tais informações em vetores de características, para então armazená-los na base de dados, o que torna possível o processamento de consultas [TORRES; FALCÃO, 2008].

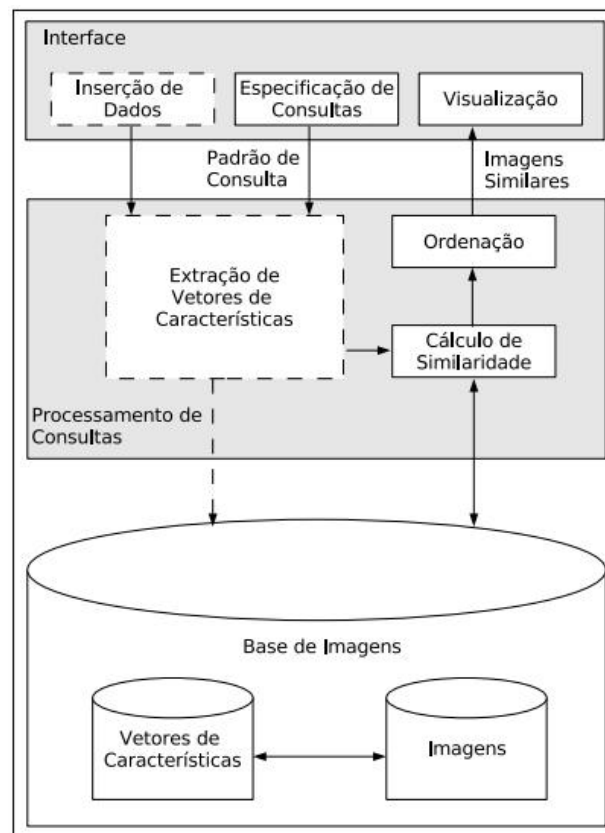


Figura 1 - Arquitetura do sistema CBIR [TORRES; FALCÃO, 2006]



Já no processamento de consultas, como pode ser visto na Figura 2, o vetor de características da imagem de consulta (o padrão de consulta) é extraído e então comparado com as imagens da base, por meio de alguma métrica de distância, por exemplo a distância Euclidiana (ver Seção 2.3, Equação 2). A partir disso, tem-se as imagens mais semelhantes ao padrão de consulta, ainda que desordenadas. Por fim, tais imagens são ordenadas de acordo com as distâncias, assim as imagens mais semelhantes em relação a uma dada imagem de consulta, aparecem nas primeiras posições do resultado [TORRES; FALCÃO, 2008].

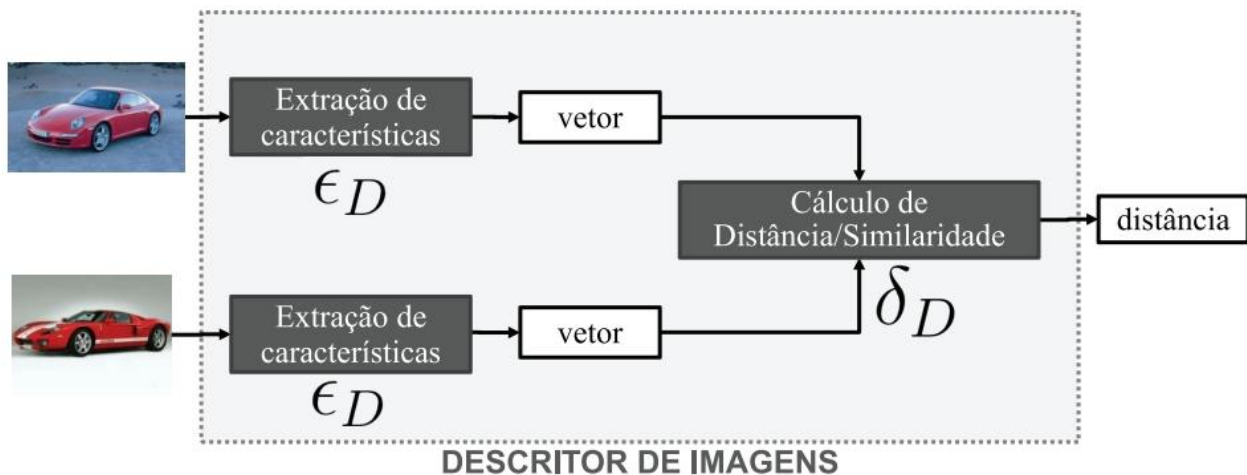


Figura 2 - Ilustração do processo de consulta [PENATTI, 2009]

### 2.3 FUNÇÕES DE DISTÂNCIA

As funções de distância (também conhecidas como métricas de dissimilaridade) tem como objetivo calcular a dissimilaridade entre as imagens da base e uma dada imagem de consulta. Para tanto, tais funções comparam os vetores de características de cada imagem e avaliam a diferença entre eles e o vetor de características da imagem de consulta [TORRES; FALCÃO, 2006]. O resultado da aplicação de uma dada função de distância é um número positivo e, quanto mais próximo de zero, mais semelhantes as imagens são, enquanto que o zero indica imagens iguais, por esse motivo utiliza-se o termo dissimilaridade.

Na literatura existem diversas propostas de funções de distâncias para aplicação nos mais diversos contextos. Dentre as principais funções de distância que são amplamente conhecidas e utilizadas, destacam-se:

- **Distância L1 (City-block):** Calcula a distância entre dois vetores de características das imagens pela acumulação das diferenças entre as características dos vetores.

$$d_1(a, b) = \|a - b\|_1 = \sum_{i=1}^d |a_i - b_i| \quad (1)$$

- **Distância L2 (distância Euclidiana):** Calcula a distância entre dois vetores de características pela soma da raiz quadrada da diferença entre  $x$  e  $y$  (no caso,  $a_i - b_i$ ).

$$d_2(a, b) = \|a - b\| = \|a - b\|_2 = \sqrt{\sum_{i=1}^d (a_i - b_i)^2} \quad (2)$$

- **Distância L $\infty$ :** Calcula a distância entre os vetores a partir da diferença máxima entre eles.

$$d_\infty(a, b) = \|a - b\|_\infty = \max_{i=1}^d |a_i - b_i| \quad (3)$$

Vale ressaltar que para todas as descrições de distância  $a$  e  $b$  são os vetores de características das imagens no espaço  $\mathbb{R}^d$  [PHILLIPS, 2014].

## 2.4 EXTRAÇÃO DE CARACTERÍSTICAS

No que diz respeito a extração de características, o processo CBIR trabalha com algoritmos extratores de características, os quais, como o próprio nome já explicita, são responsáveis pela extração das características intrínsecas da imagem, gerando como

saída os denominados vetores de características. Tais características podem ser classificadas como sendo de domínio geral e de domínio específico. Enquanto o domínio geral refere-se as características da imagem como cor, textura e forma, o domínio específico diz respeito à características que necessitam de calibração de acordo com um contexto, caso contrário dificilmente conseguiriam retornar resultados condizentes, como por exemplo classificação de impressão digital, placas de veículos, reconhecimento de faces, dentre outras, características essas dependentes da aplicação [DA SILVA; MAGALHÃES; FALCÃO, 2009].

### 2.4.1 COR

Dentre as características de domínio geral, a cor é considerada a mais importante pois é a propriedade que mais tem influência no processo de recuperação de imagens. Entretanto, para que a mesma possa ser analisada, é necessário que ela seja representada por algum sistema de cor. Dentre os mais populares, destaca-se o sistema de cor tridimensional RGB (que denota os canais *red*, *green* e *blue*) e o HSI (*hue*, *saturation* e *intensity*) [CASTAÑON, 2003].

Em relação aos extratores de características, o histograma é o extrator mais utilizado para a extração de características da cor. Como pode ser visto na Figura 3, essa técnica calcula a distribuição de cores na imagem, e gera como resultado um gráfico contendo a quantidade de pixels para determinada cor presente na imagem. Entretanto, um dos problemas encontrados nesse descritor é que, apesar do mesmo detectar cada tom da imagem, a organização especial da mesma não é levada em consideração [CASTAÑON, 2003]. Ou seja, duas imagens com histogramas iguais porém com forma e/ou textura diferentes serão consideradas como idênticas ao realizar uma comparação entre os histogramas.

Além do histograma, existe uma ampla gama de outros extratores com inúmeros refinamentos para o processo de extração de características de cor como por exemplo o *Color Correlogram* (ACC), *Border/interior pixel classification* (BIC), *Joint auto-correlogram* (JAC) e o *Color Bitmap* [DA SILVA, 2011].

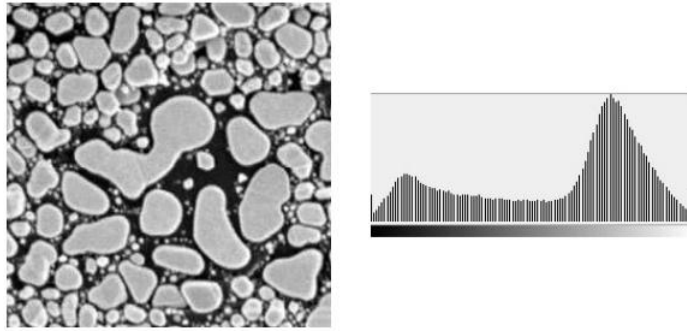


Figura 3 - Histograma de cor de uma textura [CASTAÑON, 2003]

## 2.4.2 TEXTURA

Ao contrário da cor, a textura é uma característica que não possui uma estrutura definida, já que representa apenas as formas da imagem, identificáveis pelas mudanças de intensidade na mesma, como pode ser visto na Figura 4. Por conta disso, a mesma não possui uma definição formal. Contudo, com a ajuda de descritores, é possível medir algumas de suas propriedades como a suavidade, rugosidade e regularidade [ARAUJO; OLIVEIRA, 2011].

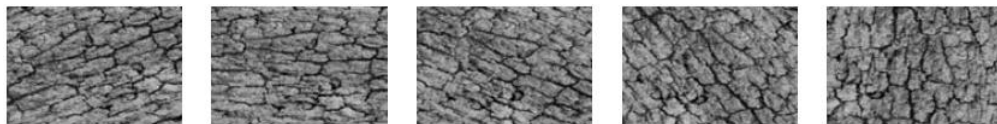


Figura 4 - Exemplos de texturas com diferentes rotações [TORRES; FALCÃO, 2008]

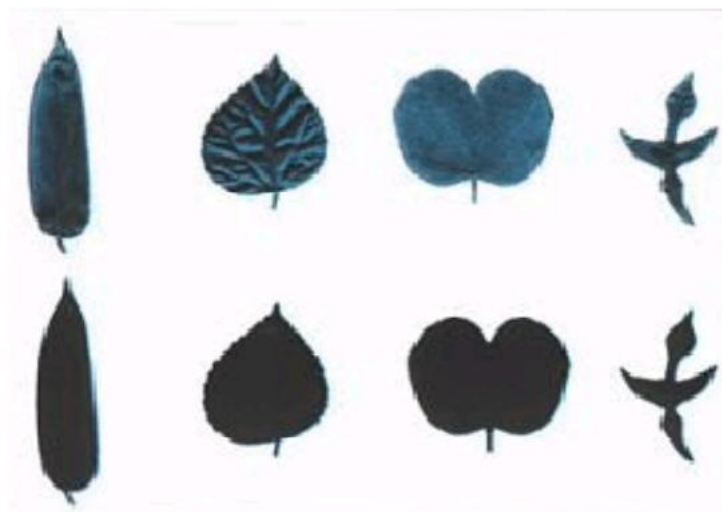
Além disso, existe na literatura três abordagens principais para adequar a recuperação de imagens por textura, que são as abordagens estruturais, estatísticas e espectrais. A abordagem estatística é a mais amplamente utilizada. Por exemplo, para que uma textura seja descrita pode-se primeiramente analisar justamente a periodicidade e angulação das ocorrências de intensidades de pixels e armazenar tais informações em estruturas denominadas como matrizes de co-ocorrência. A partir da criação de tal estrutura uma imagem é analisada para que se consiga o relacionamento entre as intensidades da mesma. Em outras palavras, a probabilidade de dois pixels cinzas ocorrerem em posições relativas [ARAUJO; OLIVEIRA, 2011].

Além de tais abordagens estatísticas, pode-se também aplicar abordagens estatísticas de primeira ordem como, por exemplo, a aplicação de medidas como média,

desvio padrão, obliquidade, curtose, entropia, dentre outras. Uma vez que espera-se que texturas similares apresentem medidas estatísticas similares.

### 2.4.3 FORMA

Como pode ser visto na Figura 5, a forma é um tipo de característica que pode representar os contornos da imagem. Apesar de ser facilmente identificável pelo olho humano, o custo computacional para recuperar imagens através dessa característica é enorme visto que, para recuperar as mesmas, é necessário conhecer as dimensões da imagem e também segmentar seus objetos [CASTAÑON, 2003].



**Figura 5 - Exemplos de formas representadas por folhas de árvores [DA FONTOURA COSTA; CESAR JR, 2000]**

Assim como na manipulação de textura, a forma tem duas abordagens demonstradas pela literatura: a abordagem baseada em contorno e baseada em região. Enquanto a primeira analisa somente os contornos da imagem, a última considera as formas da imagem como um todo. Ambas podem analisar uma forma globalmente ou em segmentos.

Contudo, a abordagem por contorno é a mais utilizada atualmente para a comparação de formas. Dentre os extratores da abordagem baseada em contorno, o *Curvature Scale Space* (CSS), por exemplo, é amplamente utilizado [PENATTI, 2009].

## 2.5 AVALIAÇÃO DE DESEMPENHO

Para avaliar o desempenho de um sistema CBIR, algumas métricas foram propostas na literatura. A medida mais utilizada para tal é a curva precisão vs. revocação [DA SILVA, 2011], onde a precisão (descrita pela Equação 1) gera como resultado a porcentagem de imagens relevantes (a partir de uma imagem de consulta) e a revocação (Equação 2) o número de imagens relevantes em relação ao número total de imagens da base de dados.

- **Equação de Precisão:**  $P(q)$  indica o número de imagens relevantes retornadas a partir da operação  $Rel(q)$  sobre o número total de imagens retornadas  $N(q)$ .

$$P(q) = \frac{Rel(q)}{N(q)} \quad (1)$$

- **Equação de Revocação:**  $R(q)$  indica o número de imagens relevantes em relação ao número total de imagens da base de dados a partir de uma consulta  $Rel(q)$  sobre a base de dados  $M(q)$ .

$$R(q) = \frac{Rel(q)}{M(q)} \quad (2)$$

Outra medida utilizada é a curva Rel x It. Essa métrica calcula o percentual de imagens relevantes trazidas pela base a cada iteração do usuário, ou seja, é possível observar como o número de imagens relevantes aumentam a cada busca efetuada [DA SILVA, 2011].

## 2.6 REALIMENTAÇÃO DE RELEVÂNCIA

Focada no feedback do usuário, a técnica de realimentação de relevância, *Relevance Feedback* (RR), é um processo interativo e iterativo que tem como objetivo solucionar um dos maiores problemas encontrados na recuperação de imagens por

conteúdo: o gap semântico. O gap semântico ocorre quando o usuário julga as imagens retornadas da consulta como irrelevantes para ele [TORRES; FALCÃO, 2008].

Além disso, tal técnica surgiu com o intuito de substituir a abordagem *computer-centric*, que exigia informações de baixo nível do usuário, como o tamanho da imagem. Como pode ser visto na Figura 6, o processo de realimentação de relevância aprimora os resultados de uma busca apenas com o feedback do usuário, ou seja, o mesmo não precisa informar características de baixo nível para a que a consulta seja realizada [RUI, 1998].

Como pode ser observado no diagrama de arquitetura, a realimentação de relevância aprende com as escolhas do usuário (imagens rotuladas como relevantes). De forma mais específica, o processo de RR identifica a cada iteração as propriedades visuais que mais caracterizam aquelas imagens, refinando cada vez mais os resultados subsequentes. Assim, o sistema CBIR traz imagens cada vez mais relevantes a cada iteração, com base na imagem de consulta informada pelo usuário. Tal processo de iterações entre o usuário e o sistema pode ser repetido até que o resultado seja alcançado, mas é preferível que termine no menor número de iterações possível.

Portanto, a técnica de realimentação de relevância tem como objetivo resolver dois aspectos: a primeira é em relação ao gap semântico entre as propriedades de alto nível (percepção visual) e de baixo nível (vetor de característica), enquanto a segunda diz respeito a percepção subjetiva da imagem pelo usuário, que pode variar com uma mesma imagem dependendo da pessoa [TORRES; FALCÃO, 2006].

No que diz respeito ao sistema de aprendizagem, existem duas abordagens para a RR: o primeiro, conhecido como paradigma guloso, é o mais utilizado no CBIR e tem como objetivo retornar as imagens mais relevantes a cada iteração com base nas escolhas do usuário. Já o segundo, paradigma planejado, especifica um número máximo de iterações e, somente após o usuário concluir todas elas, as imagens mais relevantes são retornadas [DA SILVA, 2011].

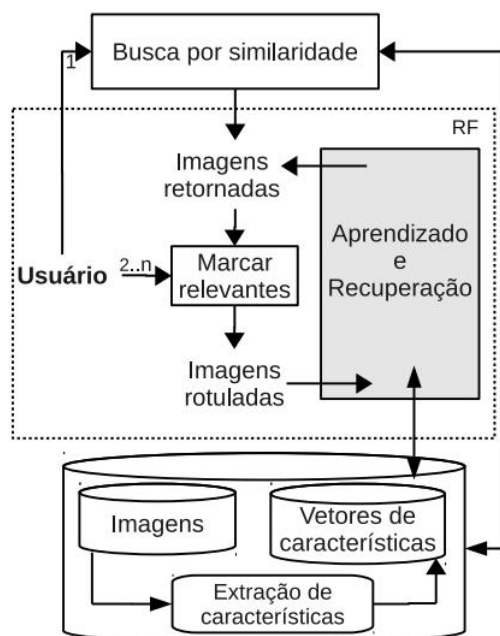


Figura 6 - Arquitetura de um Sistema CBIR com Realimentação de Relevância [DA SILVA, 2011]

### 2.6.1 MÉTODOS DE REALIMENTAÇÃO DE RELEVÂNCIA

As principais estratégias para a Realimentação de Relevância se baseiam numa “pseudo-imagem” para a realização de consultas, sendo que esta imagem pode ser usada para uma consulta ou ser utilizada em cada consulta, para posteriormente ter seus resultados da consulta fundidos.

### MOVIMENTAÇÃO DO OBJETO DE CONSULTA

Uma das primeiras técnicas criadas para RR, a Movimentação do Objeto de Consulta, *Query Point Movement* (QPM), tem como objetivo ajustar a consulta a fim de condizer o mais próximo possível das intenções do usuário. Para tanto, a mesma move os pontos de consulta relevantes para outros pontos relevantes, enquanto faz a mesma coisa com os pontos não relevantes. Ou seja, a mesma cria novos pontos de consulta a sua posição correspondente, agrupando os pontos de consulta relevantes e não-relevantes, mantendo-os distantes um do outro, o que melhora e refina cada vez mais as consultas no sistema CBIR. Na Figura 7 é possível ver uma ilustração de como essa técnica trabalha numa representação fictícia de dados [MARQUES, 2006].



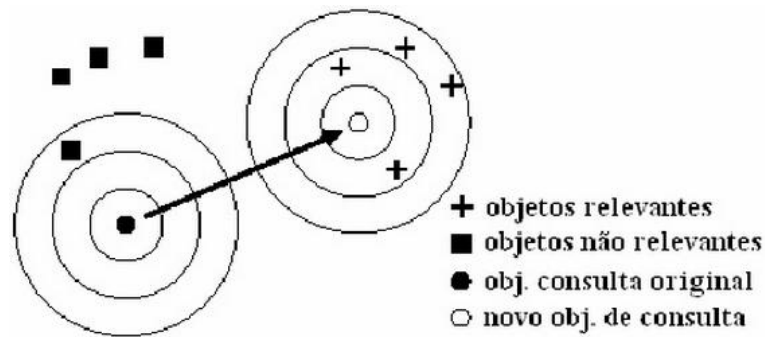


Figura 7 - Técnica de Query Point Movement utilizando a distância Euclidiana [MARQUES, 2006]

Para que se consiga comparar os resultados, essa técnica soma e normaliza os objetos relevantes e não relevantes em um novo vetor, ou seja, após mover um ponto, os vetores são organizados. A partir disso, é possível maximizar a diferença entre a média dos vetores não relevantes em relação ao vetor original da consulta, além de procurar retornar na consulta os vetores relevantes mais próximos possíveis [MARQUES, 2006]. Dentre as fórmulas utilizadas nessa técnica, destaca-se a fórmula de Rocchio, descrita formalmente pela Equação 1, onde  $Q_0$  indica o vetor original da consulta,  $Q_1$  o novo vetor da consulta,  $r$  e  $s$  o número de objetos relevantes e não relevantes e  $R_i$  e  $S_i$  indicam seus vetores, respectivamente [MARQUES, 2006].

$$Q_1 = Q_0 + \frac{1}{r} \sum_{i=1}^r R_i - \frac{1}{s} \sum_{i=1}^s S_i \quad (1)$$

Além da abordagem de Rocchio, uma outra abordagem para tal formulação foi criada por Ide [IDE, 1971]. Tal formulação modificada exclui a necessidade de normalizar os vetores e, por conta disso, gera resultados mais consistentes. Tal formulação é definida formalmente pela Equação 2.

$$Q_1 = Q_0 + \sum_{i=1}^r R_i - S_i \quad (2)$$

Por fim, é possível fazer uma pequena modificação nas fórmulas ao informar pesos (representados por  $\alpha$ ,  $\beta$  e  $\gamma$ ), o que pode aprimorar as consultas pela influência na consulta original e nos objetos, como definido formalmente pela Equação 3.

$$Q_1 = \alpha Q_0 + \frac{B}{r} \sum_{i=1}^r R_i - \frac{r}{s} \sum_{i=1}^s S_i \quad (3)$$

## REFINAMENTO DA MEDIDA DE SIMILARIDADE

Uma das maneiras de refinar uma medida de similaridade se dá pela aplicação de pesos nas funções de distância. Para tanto, cada dimensão do espaço deve possuir um peso. Com isso, é possível refinar as consultas ao dar mais foco a propriedades visuais que o usuário mais necessita para aquela consulta, como exemplo, uma determinada cor que pode ser mais importante para ele. A Equação 4 explicita um exemplo de ponderação de funções de distância, mais especificamente, funções da família Minkowski (norma  $L_p$ ) a qual abrange as distâncias Manhattan, Euclidiana e Chebychev, citadas na Seção 2.3.

$$L_p(M, Q) = \sqrt[p]{\sum_{i=0}^{k-1} w_i |M_i - Q_i|^p} \quad (4)$$

Na Equação 4,  $W_i$  refere-se aos pesos atribuídos a cada uma das características componentes do vetor de características de modo a melhor identificar as necessidades do usuário, enquanto as variáveis  $M$  e  $Q$  se referem aos atributos da consulta [MARQUES, 2006].

Contudo, existem diversas abordagens em relação a aplicação de pesos nas funções de distância. Por exemplo, o sistema MARS [PORKAEW; CHAKRABARTI, 1999] baseia seus pesos sobre o inverso do desvio padrão dos elementos relevantes, sendo que o sistema MindReader [ISHIKAWA; SUBRAMANYA; FALOUTSOS, 1998] considera a variância, ou seja, pesos proporcionais ao inverso da variância. A Figura 8 ilustra um

exemplo da modificação da abrangência de uma dada função de distância (e.g. distância Euclidiana) após a ponderação de cada uma das posições dos vetores de características, onde pode-se observar que a abrangência foi modificada de acordo com as necessidades detectadas no processo de realimentação de relevância.



Figura 8 - Exemplo de diferentes rotações possíveis que podem ser implementadas [MARQUES, 2006]

### 3 METODOLOGIA

Nesta seção serão apresentadas a base de imagens utilizada, as tecnologias, bem como a modelagem, descrição e detalhes da implementação do sistema de contexto educacional, onde foi implementado a técnica de recuperação de imagens por conteúdo em conjunto com o processo de realimentação de relevância a fim de auxiliar o aprendizado dos alunos por meio de imagens de diferentes conceitos, pela realização dos objetivos descritos na Seção 1.3.

#### 3.1 BASE DE IMAGENS

Para a realização dos experimentos no sistema proposto, foi utilizado uma base de imagens com um total de 60 imagens (com profundidade de cor de 32-bits), em que essas foram divididas em classes contendo 10 imagens cada, de acordo com o tipo da imagem. Abaixo encontra-se a descrição das classes utilizadas.

- **Base Cachorro:** Imagens de diferentes raças de cachorros.
- **Base Fruta:** Imagens de diferentes frutas (laranja, banana, abacaxi, entre outras).
- **Base Jacaré:** Imagens de diferentes jacarés.
- **Base Lagarto:** Imagens de diferentes lagartos.
- **Base Pássaro:** Imagens de diferentes pássaros.
- **Base Tigre:** Imagens de diferentes tigres.

A Figura 9 apresenta exemplos de imagens para cada uma das classes supracitadas.



Figura 9 – Exemplo de imagens para cada uma das classes da base

Tais imagens foram retiradas a partir de diversas pesquisas realizadas no motor de busca do Google [GOOGLE, 2015], em que as mesmas possuíam diferentes tamanhos e formatos. Posteriormente, para fins de padronização, as mesmas foram reduzidas para as dimensões de 150x112 pixels e convertidas para o formato JPEG com qualidade 100% pelo editor de imagens FastStone Image Viewer [FASTSTONE SOFT, 2015], para então serem armazenadas na base.

### 3.2 TECNOLOGIAS

Para o desenvolvimento do sistema, foram utilizados:

- **Microsoft Visual Studio 2013** [MICROSOFT, 2014]: Ambiente de desenvolvimento código-fechado da Microsoft que conta com recursos poderosos e suporte a diversas linguagens, foi utilizado devido a total compatibilidade com a linguagem C# e facilidade na criação da interface por meio do Windows Forms [MICROSOFT, 2015].
- **C#** [MICROSOFT, 2014]: Linguagem orientada a objetos desenvolvida pela Microsoft, simples, fortemente tipada e que roda sobre a plataforma .NET, utilizada devido a experiências anteriores com a linguagem e por possuir todos os recursos necessários para o desenvolvimento do sistema proposto.
- **Microsoft SQL Server 2008 R2** [MICROSOFT, 2014]: Sistema de Gerenciamento de Banco de Dados (SGDB) que foi escolhido devido a total integração com o Microsoft Visual Studio 2013 e por ser uma edição otimizada para base de dados pequenas. Oferece uma plataforma poderosa para gerenciar (armazenar e recuperar dados) de soluções locais e na nuvem. Disponível em várias edições.

- **FastStone Image Viewer** [FASTSTONE SOFT, 2015]: Navegador, conversor e editor de imagens que foi utilizado para reduzir o tamanho das imagens da base descrita na Seção 3.1, além de salvá-las em um formato padrão. Conta com uma variedade de recursos e suporte a diversos formatos gráficos.

### 3.3 MODELAGEM

Abaixo encontra-se a modelagem do sistema, que consistiu dos seguintes diagramas: Diagrama Entidade-Relacionamento, Modelo Entidade-Relacionamento e Diagrama de Classes.

#### 3.3.1 DIAGRAMA ENTIDADE-RELACIONAMENTO

Como pode ser visto na Figura 10, o Diagrama Entidade-Relacionamento representa o banco de dados do sistema de forma abstrata.

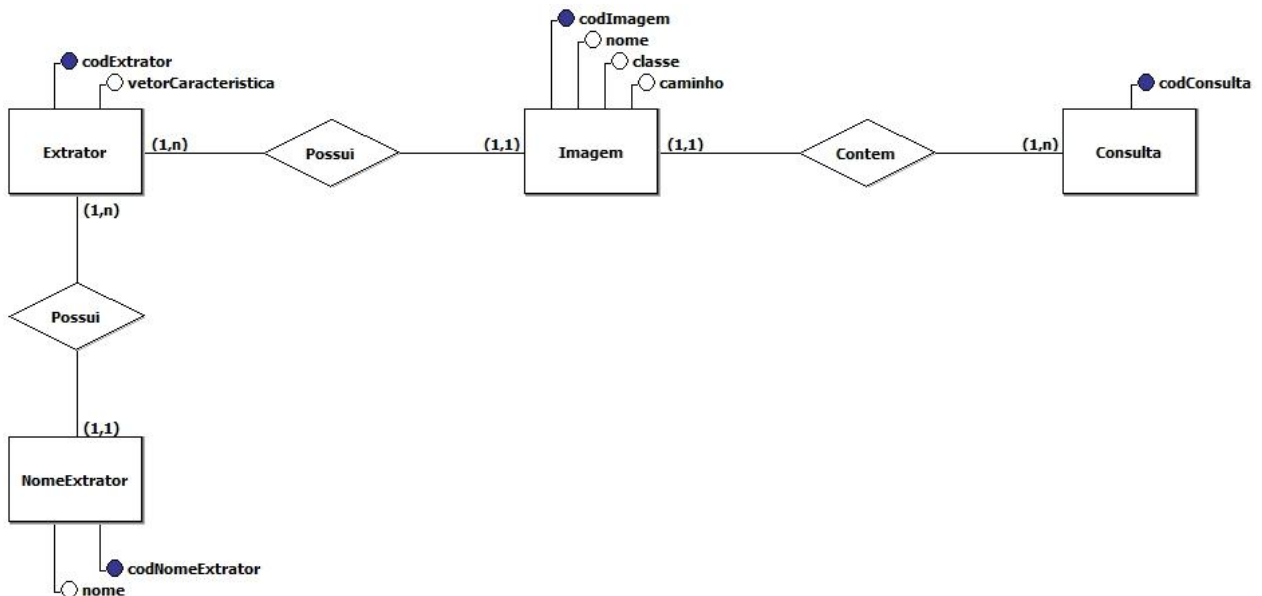


Figura 10 - Diagrama Entidade-Relacionamento

### 3.3.2 MODELO ENTIDADE-RELACIONAMENTO

O Modelo Entidade-Relacionamento apresenta a estrutura do banco de dados de maneira detalhada, como pode ser observado na Figura 11.

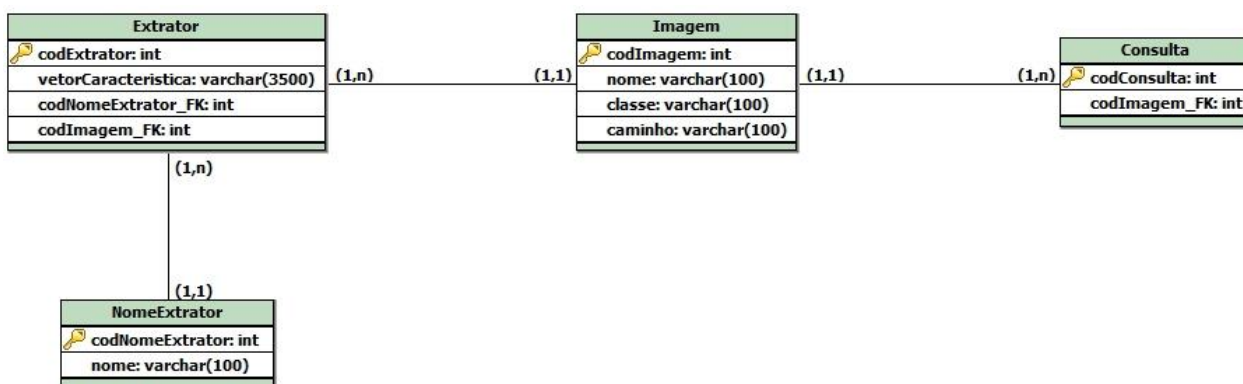


Figura 11 - Modelo Entidade-Relacionamento

Como apresentado nos diagramas acima, o banco de dados do sistema consistiu de 4 tabelas. A lista abaixo apresenta uma descrição detalhada de cada tabela do sistema e seus respectivos relacionamentos.

- **Consulta:** Armazena uma consulta de imagem realizada, contendo como chave estrangeira o atributo `codImagem_FK` da tabela `Imagem`.
- **Imagem:** Armazena informações referentes a uma imagem (nome, classe e caminho do arquivo) cadastrada no sistema.
- **Extrator:** Armazena informações referentes ao extrator da imagem, que consiste na respectiva imagem, seu vetor de características e o nome do extrator utilizado. Para tanto, relaciona-se com a tabela `Imagem` e `NomeExtrator`, contendo como chave estrangeira o atributo `codImagem_FK` da tabela `Imagem` e `codNomeExtrator_FK` da tabela `NomeExtrator`.
- **NomeExtrator:** Armazena um nome de extrator cadastrado no sistema.

### 3.3.3 DIAGRAMA DE CLASSES

O Diagrama de Classes especifica as classes do sistema, que servem de modelo para os objetos. A Figura 12 apresenta o Diagrama de Classes.

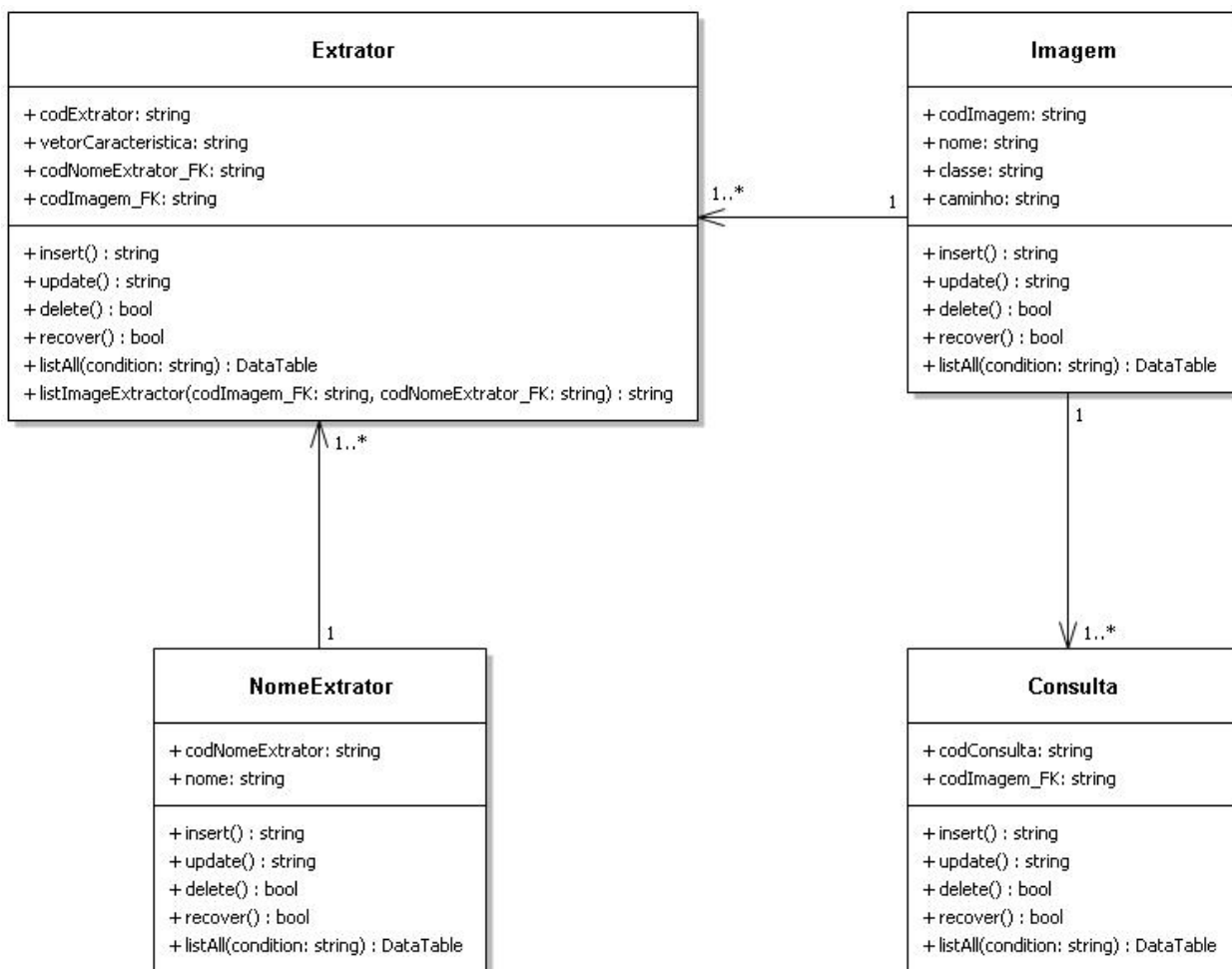


Figura 12 - Diagrama de Classes

Como pode ser visto na Figura 12, cada classe do sistema conta com métodos para a manipulação dos dados no banco. Abaixo encontra-se uma explicação detalhada de cada um dos métodos.

- **insert():** Insere uma nova tupla na base de dados, retornando como parâmetro um atributo *string* com a chave primária da inserção.
- **update():** Atualiza uma tupla na base de dados e retorna como parâmetro um atributo *string* com a quantidade de linhas afetadas.



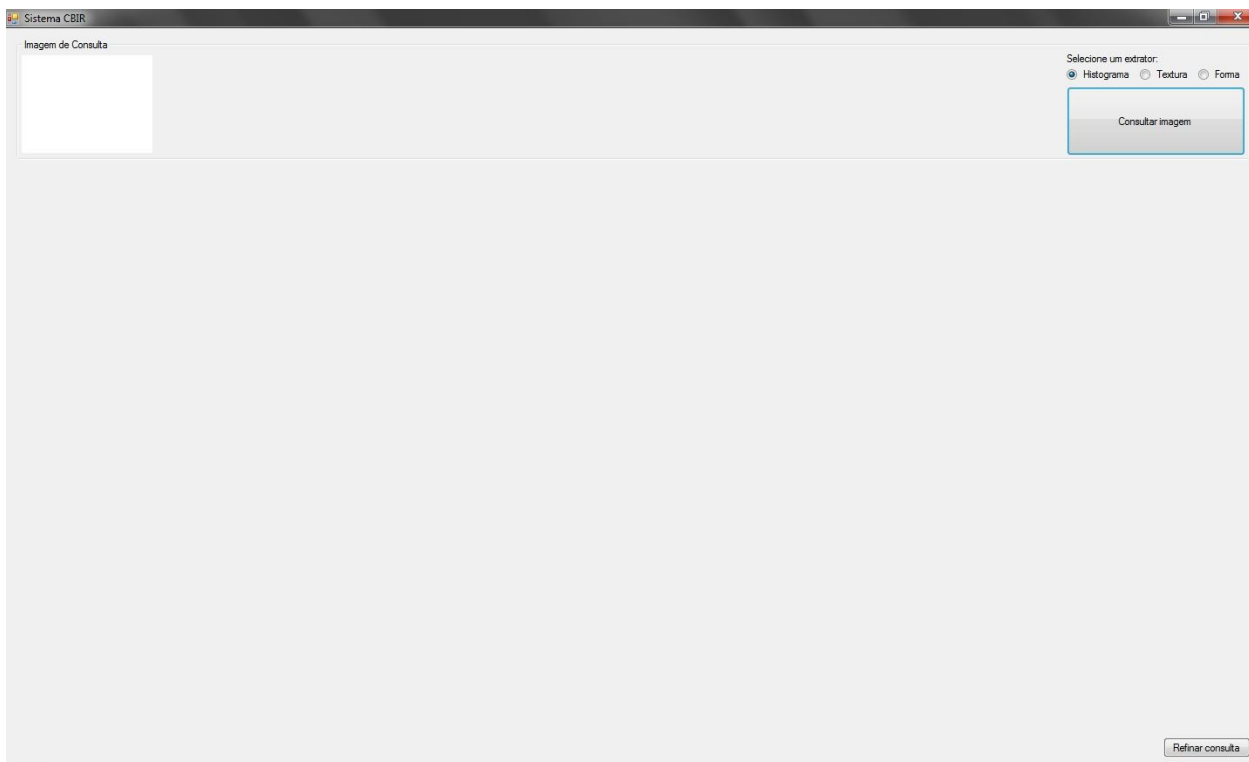
- ***delete()***: Remove uma tupla da base de dados e retorna como parâmetro um atributo bool, indicando se a remoção foi realizada com sucesso.
- ***recover()***: Recupera os atributos de uma determinada tupla para o objeto da classe, a partir da chave primária armazenada neste. Retorna como parâmetro um atributo bool, indicando se a recuperação dos dados foi realizada com sucesso.

Vale lembrar que tais métodos acima só podem ser chamados caso o objeto da classe esteja corretamente instanciado com seus respectivos atributos, já que os métodos não recebem nenhum atributo por parâmetro. Enquanto o método *insert()* necessita de um objeto já instanciado com todos os parâmetros da classe menos o código da chave primária, o método *update()* necessita de um objeto com todos os parâmetros da classe, inclusive sua chave primária. Já os métodos *delete()* e *recover()* só necessitam de um objeto instanciado com o código da chave primária, já que se tratam da exclusão/recuperação de um único elemento. Abaixo estão descritos os dois métodos de listagem criados.

- ***listAll(condition: string)***: Lista todas as tuplas existentes na base de dados da respectiva classe, retornando como parâmetro uma classe *DataTable*, que contém todas as linhas da tabela e suas respectivas colunas (atributos). Opcionalmente, o método permite passar um atributo *string* com uma condição para a cláusula WHERE, em que por padrão possui o valor "1 = 1" caso não seja informado nenhum parâmetro.
- ***listImageExtractor(codImagem\_FK: string, codNomeExtractor\_FK: string)***: Método presente na classe *Extractor* que retorna um atributo *string* com a chave primária de determinada tupla da tabela *Extractor*, com base nos valores passados por parâmetro, sendo estes obrigatórios, pois o método serve para retornar um extrator de imagem em específico.

### 3.4 SISTEMA DESENVOLVIDO

Esta seção tem como objetivo apresentar o sistema desenvolvido, descrevendo o uso geral do mesmo.



**Figura 13 – Tela principal do sistema**

Como pode ser observado na Figura 13, a tela principal do sistema possui um painel na parte superior que permite a consulta de uma imagem. Para tanto, o usuário pode selecionar um dos extratores disponíveis (histograma, textura ou forma) e então clicar no botão Consultar imagem, disponíveis na parte direita do painel. Uma nova janela se abrirá com a lista de imagens da base, onde o usuário deve escolhê-la por um clique. A Figura 14 mostra a tela de seleção da imagem de consulta.

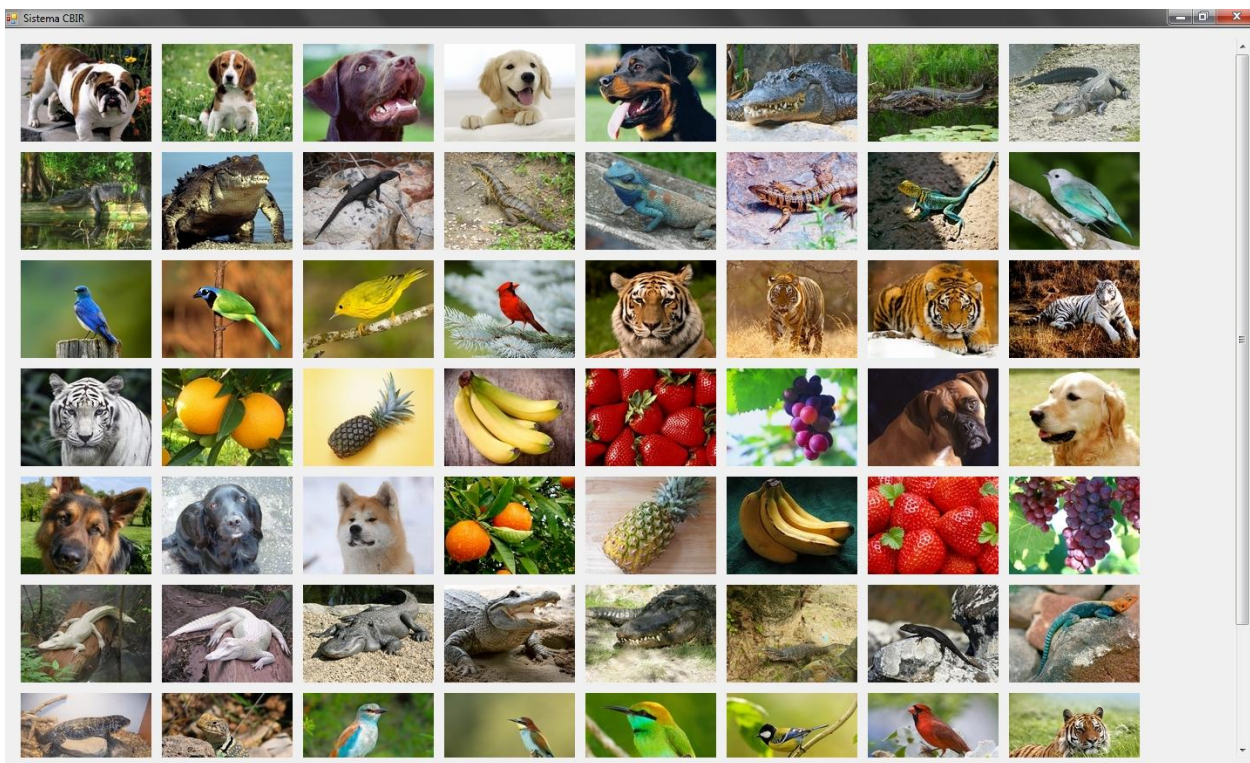
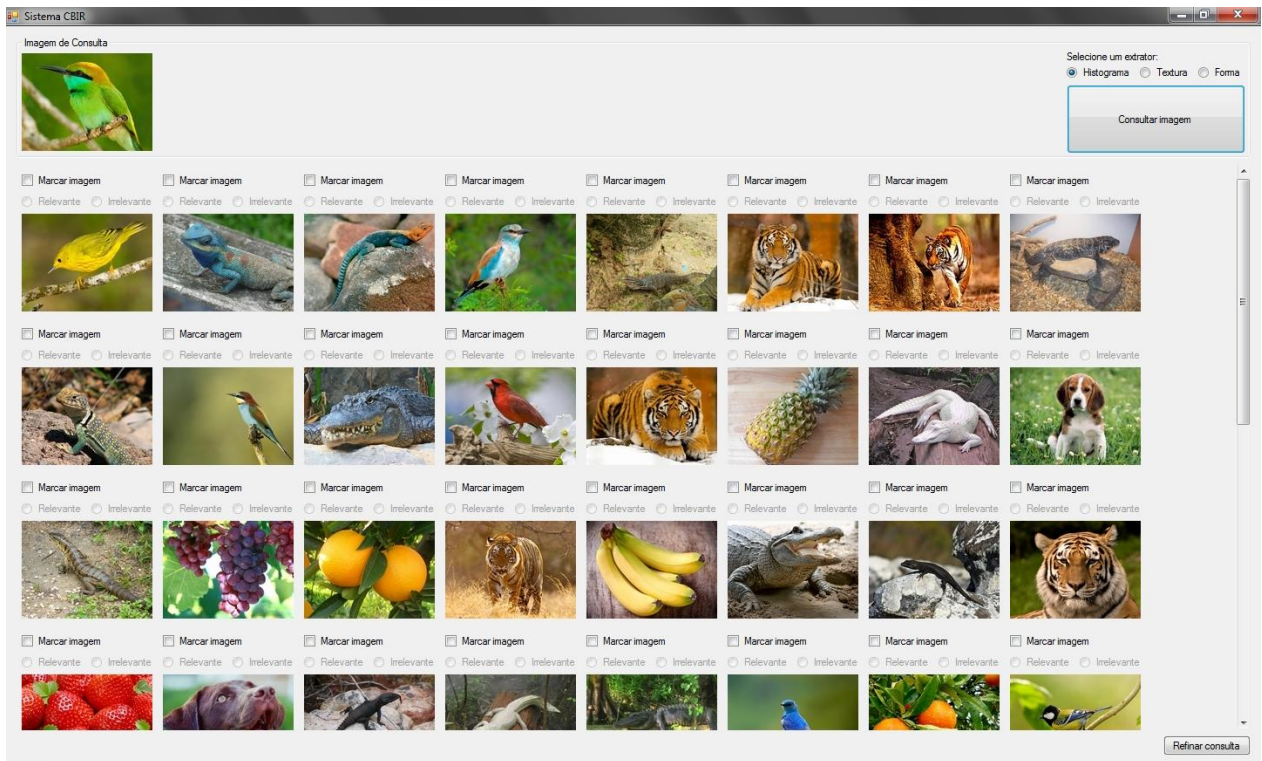


Figura 14 - Tela de seleção da imagem de consulta

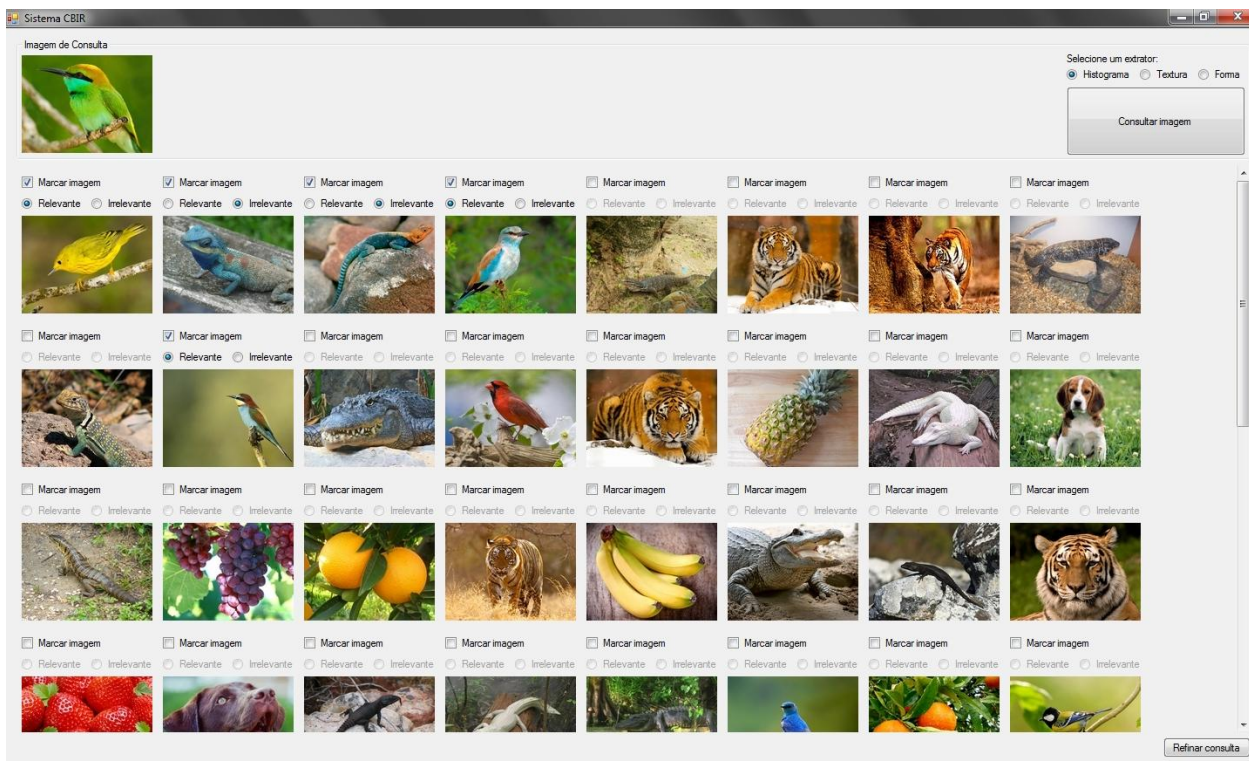
Após selecionar uma imagem, o sistema retorna para a tela principal com a lista das imagens mais relevantes para aquela consulta, baseado no extrator utilizado. A Figura 15 ilustra a tela principal após a escolha de uma imagem de consulta de um pássaro.



**Figura 15 - Tela principal após a seleção de uma imagem**

Como pode ser visto na Figura 15, o sistema implementa o método de realimentação de relevância, permitindo refinar a consulta após o resultado inicial. Para tanto, o usuário deve selecionar as imagens como relevante/irrelevante, marcando a caixa de seleção “Marcar imagem” e então selecionando algum dos botões de rádio “Relevante” ou “Irrelevante”. Após o julgamento das imagens, é possível clicar no botão “Refinar Consulta”, que traz um novo resultado mais preciso com as indicações do usuário. Esse processo pode se repetir indefinidamente, até que o usuário esteja satisfeito com o resultado. A Figura 16 ilustra o processo de seleção das imagens relevantes/irrelevantes.





**Figura 16 - Seleção das imagens relevantes/irrelevantes**

Após a seleção das imagens relevantes/irrelevantes, a consulta é refinada em uma iteração, trazendo um resultado mais condizente com as indicações do usuário, como mostra a Figura 17.

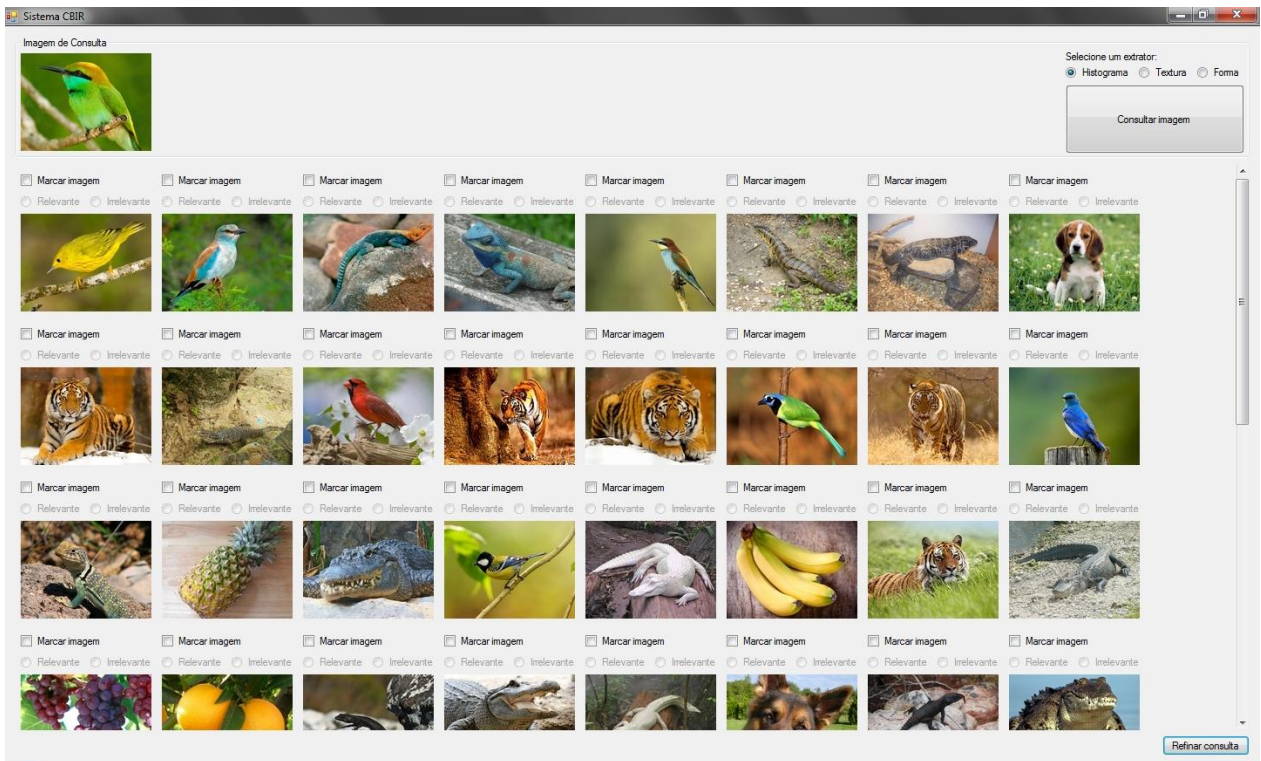


Figura 17 - Consulta refinada após a seleção das imagens relevantes/irrelevantes

### 3.5 IMPLEMENTAÇÃO

Nesta seção, serão apresentados detalhes sobre o desenvolvimento do sistema proposto tais como a maneira que foram extraídos os vetores de características das imagens e como o cálculo de distância e o algoritmo de Rocchio foram implementados.

Primeiramente, para permitir a escolha dos três extratores disponíveis (histograma, textura e forma) em uma consulta, foi necessário extrair para cada imagem da base descrita na Seção 3.1 os seus respectivos vetores de características, com cada um dos extratores supracitados. Para tanto, foi utilizado um aplicativo disponibilizado pelo orientador que permitiu extrair (utilizando tais extratores) os vetores de características em arquivos texto, onde estes foram posteriormente inseridos na tabela Extrator do banco de dados.

Em relação ao cálculo de distância, para todos os extratores foi utilizado a distância Euclidiana (ver Seção 2.3, Equação 2). Assim, como pode ser visto no Quadro 1, foram criados os métodos *calculateDistance()*, *addDistance(string codExtrator, string codImagem\_FK, double distancia)* e *sortDistances()*, onde *IstDistances* se trata da lista

de distâncias atual, *lstVectors* a lista dos vetores de características, *objVector* um dado vetor de característica da lista *lstVectors* e *objVectorQuery* o vetor de característica da imagem de consulta. As variáveis *sum* e *a* servem para armazenar o cálculo das diferenças durante a execução do algoritmo.

```
private void calculateDistance()
{
    lstDistances.Clear();

    foreach (VetorCaracteristica objVector in lstVectors)
    {
        if (objVectorQuery.pCodImagem_FK == objVector.pCodImagem_FK)
        {
            continue;
        }

        double sum = 0, a = 0;
        for (int i = 0; i < objVectorQuery.pCaracteristica.Count; i++)
        {
            a = (double.Parse(objVectorQuery.pCaracteristica[i]) -
double.Parse(objVector.pCaracteristica[i]));
            if (a < 0)
            {
                a = a * (-1);
            }
            sum += a;
        }
        addDistance(objVector.pCodExtrator, objVector.pCodImagem_FK, sum);
    }
}

private void addDistance(string codExtrator, string codImagem_FK, double distancia)
{
    Distancia objDistance = new Distancia();
    objDistance.pCodExtrator = codExtrator;
    objDistance.pCodImagem_FK = codImagem_FK;
    objDistance.pDistancia = distancia;
    lstDistances.Add(objDistance);
}

private void sortDistances()
{
    lstDistances = lstDistances.OrderBy(x => x.pDistancia).ToList();
}
}
```

**Quadro 1 - Algoritmo de cálculo de distância**

Como pode ser observado no Quadro 1, logo no início da execução é realizada uma exclusão de todos os elementos previamente existentes na lista *lstDistances*, pela chamada do método *Clear()*. Tal ação é necessária pois esse método é chamado diversas vezes para calcular novas distâncias. Após isso, itera-se por todos os vetores de características da lista *lstVectors* para o cálculo de diferença entre o vetor da imagem de consulta e os vetores das imagens da base. Uma comparação inicial verifica se a imagem de consulta é igual à imagem que está sendo comparada no momento: caso sim, é executado o comando *continue*, que faz com que a iteração atual seja ignorada,

passando direto para a próxima. Então, um laço de repetição é feito por todas as características do vetor da imagem de consulta e de um dado vetor da lista *IstVectors* para que seja feito o cálculo de subtração sobre cada uma das características e posterior somatório, em que, como trata-se de uma subtração, é verificado antes se o resultado da variável *a* é menor que 0, caso sim, o valor é multiplicado por -1 para que se torne um valor positivo e possa ser adicionado a variável *sum*. Tem-se como resultado o somatório final (variável *sum*) que é a distância (diferença) entre a imagem de consulta original e a imagem comparada. O método *addDistance(string codExtractor, string codImagem\_FK, double distancia)* é então chamado com seus devidos parâmetros, que adiciona um novo objeto distância para a lista *IstDistances*. Com a lista de distâncias preenchida, chama-se o método *sortDistances()*, que ordena a lista de distâncias por ordem de menor para maior.

O algoritmo de Rocchio foi implementado com a possibilidade de marcar tanto imagens relevantes quanto irrelevantes, sendo que foram atribuídos os pesos de 0.75 para beta (imagens relevantes) e 0.25 para gama (imagens irrelevantes), pois normalmente aplica-se um peso maior para as imagens relevantes. Assim, foi criado o método *rochioAlgorithm()*, onde a variável *sum* serve para armazenar temporariamente o somatório das características, a variável *b* refere-se ao peso das imagens relevantes e a variável *c* ao peso das imagens irrelevantes. As listas *IstRelevantImages* e *IstIrrelevantImages* armazenam todos os vetores de características das imagens que o usuário julgou como relevante e irrelevante, respectivamente, enquanto as listas *IstRelevantVector* e *IstIrrelevantVector* armazenam um único vetor de característica, que terá seus valores a partir do somatório feito sobre todas as características de cada imagem presentes nas respectivas listas de vetores supracitadas, além do posterior cálculo da média e aplicação do peso. O *IstResultVector* se trata do vetor de característica final, que tem seu resultado após a soma do vetor da imagem de consulta com o vetor relevante (*IstRelevantVector*) e por fim da subtração do vetor resultado (*IstResultVector*) sobre o vetor irrelevante (*IstIrrelevantVector*), se esse existir. Tem-se então o *objVectorQuery* que trata-se do vetor de característica da imagem de consulta, enquanto o *objVector* é um dado vetor de características da lista *IstRelevantImages* ou *IstIrrelevantImages*. O Quadro 2 apresenta o algoritmo de Rocchio implementado.



```

private List<string> rocchioAlgorithm()
{
    double sum = 0, b = 0.75, c = 0.25;
    List<double> lstRelevantVector = new List<double>();
    List<double> lstIrrelevantVector = new List<double>();
    List<double> lstResultVector = new List<double>();

    // Calculo do vetor de caracteristica relevante/irrelevante
    for (int i = 0; i < objVectorQuery.pCaracteristica.Count; i++)
    {
        foreach (VetorCaracteristica objVector in lstRelevantImages)
        {
            sum += double.Parse(objVector.pCaracteristica[i]);
        }
        sum = (sum / lstRelevantImages.Count) * b;
        lstRelevantVector.Add(sum);
        sum = 0;

        if (lstIrrelevantImages.Count != 0)
        {
            foreach (VetorCaracteristica objVector in lstIrrelevantImages)
            {
                sum += double.Parse(objVector.pCaracteristica[i]);
            }
            sum = (sum / lstIrrelevantImages.Count) * c;
            lstIrrelevantVector.Add(sum);
            sum = 0;
        }
    }

    // Calculo final (vetor da imagem de consulta + vetor das imagens relevantes - vetor das
    imagens irrelevantes)
    for (int i = 0; i < objVectorQuery.pCaracteristica.Count; i++)
    {
        sum = (double.Parse(objVectorQuery.pCaracteristica[i]) + lstRelevantVector[i]);
        lstResultVector.Add(sum);
    }

    if (lstIrrelevantImages.Count != 0)
    {
        for (int i = 0; i < objVectorQuery.pCaracteristica.Count; i++)
        {
            lstResultVector[i] = (lstResultVector[i] - lstIrrelevantVector[i]);
            if (lstResultVector[i] < 0)
            {
                lstResultVector[i] = 0;
            }
        }
    }

    return lstResultVector.Select(x => x.ToString()).ToList();
}

```

**Quadro 2 - Algoritmo de Rocchio**

Como é possível observar no Quadro 2, a primeira seção do algoritmo executa um laço de repetição percorrendo todas as características, onde dentro deste itera-se sobre todos os vetores de imagens relevantes da lista *lstRelevantImages*. Assim, é feito um somatório sobre uma dada característica dos vetores relevantes, em que no final do laço interior tem-se como resultado o somatório final de uma característica, extraído de todos os vetores relevantes. Este somatório é então dividido pela quantidade de elementos da lista *lstRelevantImages* para que se obtenha a média e, a seguir, é aplicado

seu respectivo peso pela multiplicação do resultado com as variáveis *b* ou *c*. Com o cálculo efetuado armazenado na variável *sum*, adiciona-se o resultado ao vetor relevante *IstRelevantVector*, criando uma nova característica, sendo que este processo se repete até a iteração de todas as características. O segundo laço de repetição, após checar se de fato existem imagens irrelevantes, realiza o mesmo procedimento porém para os vetores de imagens irrelevantes. Desta forma, ao final de todas as iterações do bloco de código inicial tem-se como resultado o vetor de características relevante *IstRelevantVector* e o vetor de características irrelevante *IstIrrelevantVector*, que serão usados no cálculo final.

Com os dois vetores preenchidos é realizado o cálculo final do vetor, onde um laço de repetição percorre cada característica do vetor da imagem de consulta (*objVectorQuery*) e é feita uma soma com a característica equivalente do vetor relevante *IstRelevantVector*. Essa soma é armazenada temporariamente na variável *sum*, em que então é adicionada ao vetor resultado *IstResultVector*, criando assim uma nova característica. Terminada a soma com o vetor original e o vetor relevante, é checado novamente se existem imagens irrelevantes: caso sim, executa-se o bloco final do código, que percorre em um laço de repetição todas as características do vetor resultado *IstResultVector*, subtraindo-se com a característica equivalente do vetor irrelevante *IstIrrelevantVector*. Ainda dentro do laço é feita uma comparação para verificar se o resultado da subtração foi menor que 0, se sim a característica em questão recebe o valor 0 já que termos negativos no algoritmo de Rocchio são transformados para 0. Tendo o vetor de características resultado *IstResultVector* agora calculado, o mesmo é retornado pelo método, finalizando a execução do algoritmo.

### 3.6 CONSIDERAÇÕES FINAIS

Esta seção apresentou detalhes da implementação do sistema proposto, onde nas seções 3.1, 3.2 e 3.3 foram descritas a base de imagens utilizada, suas tecnologias e a modelagem geral do mesmo, que incluiu o Diagrama Entidade-Relacionamento, Modelo Entidade-Relacionamento e por fim o Diagrama de Classes.

Já na Seção 3.4 foi apresentado o sistema desenvolvido, contendo uma explicação geral do uso do mesmo por meio de telas da interface.

Com relação ao desenvolvimento do sistema, foi utilizado a distância Euclidiana (Seção 2.3, Equação 2) para o cálculo de distância e o algoritmo de Rocchio com aplicação de pesos (Seção 2.6.1, Equação 3) para o método de realimentação de relevância, onde a implementação destes foram explicadas em detalhes na Seção 3.5, sendo que tais métodos foram aplicados sobre a base de imagens descrita na Seção 3.1 com os extratores de histograma, textura e forma.

Dessa forma, o sistema implementado teve como objetivo permitir a realização de análises dos resultados em tempo real, para que estes posteriormente possam ser calibrados para o contexto educacional.

## 4 RESULTADOS E DISCUSSÕES

Nesta seção serão apresentados os experimentos realizados sobre a base de imagens descrita na Seção 3.1, onde para a realização dos testes foram utilizados a distância Euclidiana (Seção 2.3, Equação 2) para o cálculo de distância e o algoritmo de Rocchio com aplicação de pesos (Seção 2.6.1, Equação 3) para o método de realimentação de relevância, sobre os extratores de histograma, textura e forma.

Tais experimentos foram realizados por meio de testes de precisão (Seção 2.5, Equação 1) e teve como objetivo determinar quais dos extratores utilizados trouxeram o melhor resultado no cenário supracitado, além de verificar a influência do processo de realimentação de relevância na consulta original, para que posteriormente possam ser calibrados para o sistema de contexto educacional.

### 4.1 TESTES DE PRECISÃO

Para a realização dos testes de precisão, foram utilizados os pesos de 0.75 para beta (imagens relevantes) e 0.25 para gama (imagens irrelevantes) no algoritmo de Rocchio com os extratores supracitados.

Assim, foram utilizadas 5 imagens, em que para cada imagem foram realizadas consultas sobre cada um dos extratores, além de duas iterações sobre a mesma (consulta inicial e consulta refinada). A Figura 18 apresenta as imagens utilizadas nos testes de precisão.



**Figura 18 – Imagem de um pássaro (1); Imagem de um tigre (2); Imagem de um lagarto (3); Imagem de um jacaré (4); Imagem de um cachorro (5)**

Abaixo encontra-se o resultado dos testes de precisão realizado sobre as imagens.

**Tabela 1 - Testes de precisão sobre a imagem 1**

Imagem 1		
Extrator	Consulta inicial	Consulta refinada
Histograma	40%	50%
Textura	40%	20%
Forma	10%	40%

**Tabela 2 - Testes de precisão sobre a imagem 2**

Imagem 2		
Extrator	Consulta inicial	Consulta refinada
Histograma	30%	40%
Textura	30%	50%
Forma	0%	30%

**Tabela 3 - Testes de precisão sobre a imagem 3**

Imagem 3		
Extrator	Consulta inicial	Consulta refinada
Histograma	30%	50%
Textura	10%	0%
Forma	0%	40%

**Tabela 4 - Testes de precisão sobre a imagem 4**

Imagem 4		
Extrator	Consulta inicial	Consulta refinada
Histograma	60%	60%
Textura	50%	40%
Forma	10%	40%

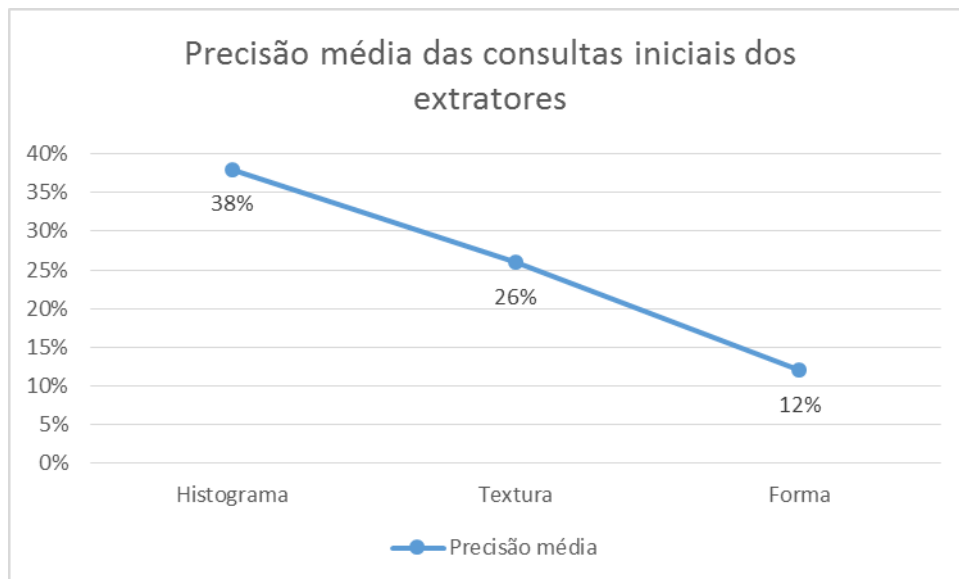
**Tabela 5 - Testes de precisão sobre a imagem 5**

Imagem 5		
Extrator	Consulta inicial	Consulta refinada
Histograma	30%	30%
Textura	0%	10%
Forma	40%	40%

## 4.2 RESULTADOS OBTIDOS

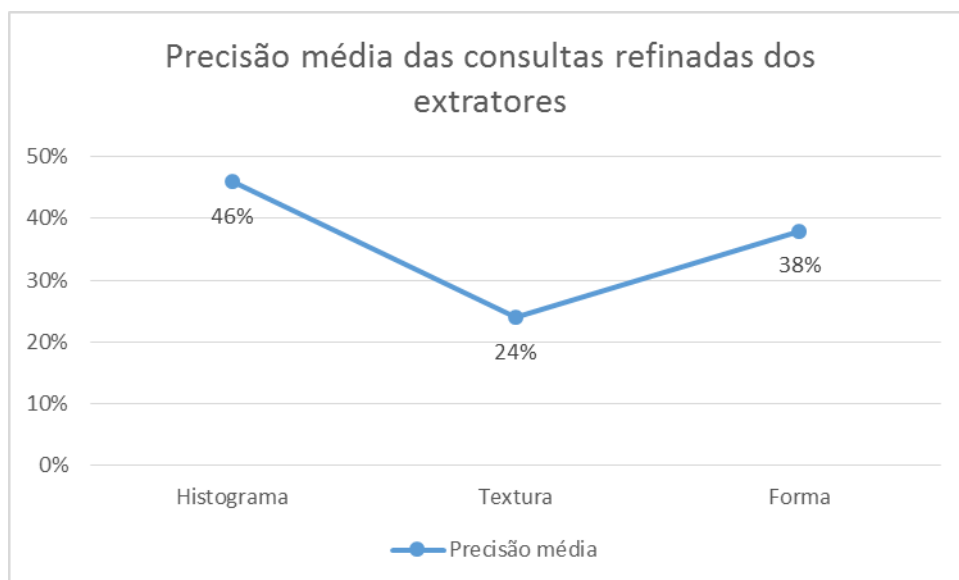
Como pode ser observado na Seção 4.1, com base nos testes de precisão realizados obteve-se como resultado uma precisão média de 38% para o extrator de histograma, enquanto os extratores de textura e forma tiveram 26% e 12%,

respectivamente. O Gráfico 1 ilustra os resultados de precisão média para as consultas iniciais dos extratores.



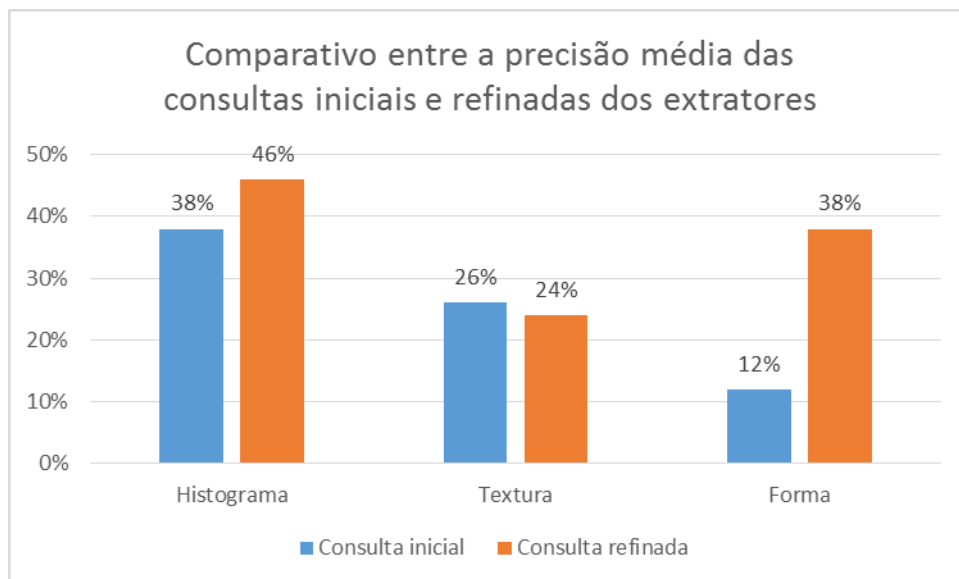
**Gráfico 1 - Gráfico de precisão média para as consultas iniciais dos extratores**

Com relação aos testes de precisão realizados sobre a consulta refinada, teve-se como resultado uma precisão média de 46% para o extrator de histograma, enquanto obteve-se 24% e 38% para os extratores de textura e forma, respectivamente. O Gráfico 2 ilustra os resultados de precisão média para as consultas refinadas dos extratores.



**Gráfico 2 - Gráfico de precisão média para as consultas refinadas dos extratores**

Por fim, com o objetivo de verificar a influência do processo de realimentação de relevância no sistema proposto, foi feito um comparativo entre a precisão média das consultas iniciais e consultas refinadas dos respectivos extratores. O Gráfico 3 ilustra o comparativo realizado entre a precisão média das consultas iniciais e refinadas dos extratores.



**Gráfico 3 - Comparativo entre a precisão média das consultas iniciais e refinadas dos extratores**

### 4.3 CONSIDERAÇÕES FINAIS

Esta seção apresentou os experimentos realizados sobre o sistema proposto, em que tinha-se como objetivo principal verificar quais dos extratores trouxeram os melhores resultados, além de verificar a influência do processo de realimentação de relevância sobre a consulta inicial.

Com base nos testes realizados, pôde-se concluir que na consulta inicial o extrator de histograma trouxe os melhores resultados nas imagens testadas, com uma precisão média de 38%, sendo que o extrator de textura veio em segundo lugar com 26% e por último o extrator de forma com 12%.

Já em relação a consulta refinada, que denota a consulta após uma iteração do processo de realimentação de relevância (seleção de imagens relevantes/irrelevantes), o extrator de histograma também obteve os melhores resultados sobre as imagens

testadas, com uma precisão média de 46% em comparação aos 38% e 24% dos extratores de forma e textura, respectivamente.

Com relação a influência do processo de realimentação de relevância, o extrator de histograma obteve uma melhora de 38% para 46% em comparação com a consulta inicial, enquanto o extrator de textura teve uma diferença irrelevante de 26% para 24%. Por outro lado, o extrator de forma obteve uma melhora considerável sobre a consulta inicial, de 12% para 38%.

Dessa forma, concluiu-se a importância de se testar diferentes extratores além de verificar a influência do processo de realimentação de relevância para o contexto em questão, onde pôde-se concluir que o extrator de histograma trouxe os melhores resultados para todos os casos, e o processo de realimentação de relevância teve, no geral, uma influência positiva sobre os resultados pelo refino da consulta inicial.



## 5 CONCLUSÃO

Este trabalho possibilitou a realização dos objetivos listados na Seção 1.3, que tinha como objetivo principal o estudo e análise dos métodos de realimentação de relevância para que fossem posteriormente calibrados em um sistema de recuperação de imagens de contexto educacional.

Como resultado, foi gerado um sistema que permitia a escolha de três extratores diferentes, com o algoritmo de Rocchio implementado, um método de realimentação de relevância. Tal sistema permitiu a análise dos resultados em tempo real, em que foram testados diferentes extratores além de se verificar a influência do processo de realimentação de relevância, onde foi constatado qual foi o melhor extrator para o contexto em questão, além da influência positiva do processo de realimentação de relevância sobre as consultas.

Dessa forma, foi possível validar a eficácia dessa abordagem para o ensino de conceitos por meio de imagens, pelo desenvolvimento do sistema proposto e posterior análise dos resultados obtidos nos testes realizados.

### 5.1 TRABALHOS FUTUROS

Como sugestões futuras para aprimorar o sistema desenvolvido, pode-se citar a utilização de outras medidas de similaridade, em que dependendo dos casos podem trazer maior precisão nos resultados e um melhor desempenho na consulta, ou até mesmo trazer um maior desempenho mas menor precisão nos resultados, e vice-versa. Como exemplo, tem-se funções de distância específicas para a comparação de histogramas como a função de distância de intersecção de histogramas, que teoricamente é capaz de ignorar as influências irrelevantes do fundo da imagem, e a função de distância  $d_{Log}$ , que pode aumentar a precisão pela diminuição do efeito de bins com valores muito altos no resultado final do cálculo de distância [PENATTI, 2009]. Além das diferentes medidas de similaridade, como pode ser visto na Seção 2.6.1, Equação 4, é possível aplicar pesos as funções de distância da família Minkowski (norma

Lp), o que permite dar maior relevância a certas propriedades visuais, como por exemplo uma cor em específico, dessa forma refinando a função de distância.

Outra forma de aprimorar a consulta é pelo uso de outros descritores (responsáveis pela extração das características de uma imagem) mais específicos para as características de cor, textura e forma. Hoje em dia existem diversos descritores para cada uma das características supracitadas, sendo que algumas dessas estão citadas nas seções 2.4.1 e 2.4.3, respectivamente.

Além disso, como pode ser observado na Seção 2.6.1, Equação 2, existem também outras variações do algoritmo de Rocchio que podem ser aplicadas, como por exemplo a abordagem criada por Ide [IDE, 1971] que exclui a necessidade de normalizar os vetores previamente e, com isso, pode gerar resultados mais consistentes.

## REFERÊNCIAS

ARAUJO, Lucas Moreno; OLIVEIRA, Fernando Luiz. **Estudo dos descritores de características primitivas**. Centro Universitário Luterano de Palmas. In: ENCONTRO DE COMPUTAÇÃO E INFORMÁTICA DO TOCANTINS, 13., 2011, Palmas. Anais... Palmas: CEULP/ULBRA, 2011. p. 169-178. Disponível em <[http://ulbra-to.br/encoinfo/artigos/2011/Estudo\\_dos\\_descritores\\_de\\_caracteristicas\\_primitivas.pdf](http://ulbra-to.br/encoinfo/artigos/2011/Estudo_dos_descritores_de_caracteristicas_primitivas.pdf)>. Acesso em: 23 nov. 2014.

CARNEIRO, Liliane Bernardes; MANINI, Miriam Paula. Alfabetização visual - desafio para o profissional da informação no fomento à leitura de imagens de livros infantis. In: CONGRESO ISKO-ESPAÑA, 4., 2009, Valência. **Nuevas perspectivas para la difusión y organización del conocimiento**. Valência: Editorial UPV, 2009. p. 381-397. Disponível em: <<http://dialnet.unirioja.es/descarga/articulo/2923210.pdf>>. Acesso em: 22 nov. 2015.

CASTAÑÓN, Cesar Armando Beltran. **Recuperação de imagens por conteúdo através de análise multiresolução por Wavelets**. 2003. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2003. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-29072004-194807>>. Acesso em: 23 nov. 2014.

DA FONTOURA COSTA, Luciano; CESAR JR, Roberto Marcond. **Shape analysis and classification: theory and practice**. CRC press, 2000.

DA SILVA, André Tavares. **Recuperação de imagens por conteúdo baseada em realimentação de relevância e classificador por floresta de caminhos ótimos**. 2011. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2011.

DA SILVA, André Tavares; MAGALHÃES, Léo Pini; FALCÃO, Alexandre Xavier. **Recuperação de Imagens por Realimentação de Relevância**. In: EADCA - Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial, 2, 2009, Campinas. Anais... São Paulo: Universidade de Campinas – UNICAMP, 2009.

DONDIS, Donis. A. **Sintaxe da linguagem visual**. São Paulo: Martins Fontes, 1997.

DUBORGEL, Bruno. **Imaginário e Pedagogia**. Lisboa: Instituto Piaget, 1992.

FASTSTONE                      SOFT.                      **FastStone**.                      Disponível                      em:  
<<http://www.faststone.org/FSViewerDetail.htm>>. Acesso em: 07 set. 2015.

GOOGLE. **GOOGLE IMAGES**. Disponível em: <<http://www.google.com/imghp>>. Acesso em: 16 nov. 2015.

GUDIVADA, Venkat N.; RAGHAVAN, Vijay V. **Content based image retrieval systems**. Computer, v. 28, n. 9, p. 18-22, 1995.

IDE, Eleanor. New experiments in relevance feedback. **The SMART retrieval system**, p. 337-354, 1971.

ISHIKAWA, Yoshiharu; SUBRAMANYA, Ravishankar; FALOUTSOS, Christos. **MindReader: Querying databases through multiple examples**. Computer Science Department, p. 551, 1998.

LENCASTRE, José Alberto; CHAVES, José Henrique. Ensinar pela imagem. **Revista Galego-portuguesa de Psicoloxía e Educación**, [s. L.], v. 10, n. 8, p. 2100-2105, jul. 2003. Disponível em: <[http://repositorium.sdum.uminho.pt/bitstream/1822/26021/1/Lencastre\\_ENSINAR\\_PEL\\_A\\_IMAGEM\\_2003.pdf](http://repositorium.sdum.uminho.pt/bitstream/1822/26021/1/Lencastre_ENSINAR_PEL_A_IMAGEM_2003.pdf)>. Acesso em: 22 nov. 2015.

LENCASTRE, José Alberto; CHAVES, José Henrique. A imagem como linguagem. In: BARCA, A., PERALBO, M., PORTO, A., DUARTE DA SILVA, B. E ALMEIDA, L. (EDS.), 2007, A Corunha. **Libro de Actas do Congreso Internacional Galego-Portugués de Psicopedagogía**. A Corunha: Universidade da Coruña, 2007. p. 1162-1173. Disponível em: <[http://repositorium.sdum.uminho.pt/bitstream/1822/26093/1/Lencastre\\_A\\_imagem\\_co\\_mo\\_linguagem\\_Galaico2007.pdf](http://repositorium.sdum.uminho.pt/bitstream/1822/26093/1/Lencastre_A_imagem_co_mo_linguagem_Galaico2007.pdf)>. Acesso em: 22 nov. 2015.

MARQUES, Joselene. **Realimentação de relevância para recuperação por conteúdo de imagens médicas visando diminuir a descontinuidade semântica**. 2006. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2006. Disponível em: <[http://www.academia.edu/452934/Realimentação\\_de\\_relevância\\_para\\_recuperação\\_por\\_conteúdo\\_de\\_imagens\\_médicas\\_visando\\_diminuir\\_a\\_descontinuidade\\_semântica](http://www.academia.edu/452934/Realimentação_de_relevância_para_recuperação_por_conteúdo_de_imagens_médicas_visando_diminuir_a_descontinuidade_semântica)>. Acesso em: 23 nov. 2014.

MICROSOFT. **Visual C#**. Disponível em: <<http://msdn.microsoft.com/en-us/library/kx37x362.aspx>>. Acesso em: 08 dez. 2014.

MICROSOFT. **Visual Studio**. Disponível em: <<http://www.visualstudio.com>>. Acesso em: 08 dez. 2014.

MICROSOFT. **SQL Server**. Disponível em: <<http://www.microsoft.com/en-us/server-cloud/products/sql-server/default.aspx>>. Acesso em: 09 dez. 2014.

MICROSOFT. **Windows Forms**. Disponível em: <[http://msdn.microsoft.com/en-us/library/dd30h2yb\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/dd30h2yb(v=vs.110).aspx)>. Acesso em: 15 nov. 2015.

MODERNO, António. **A Comunicação Audiovisual no Processo Didático: no Ensino, na Formação Profissional**. Aveiro: Universidade de Aveiro, 1992.

PENATTI, Otavio Augusto Bizetto. **Estudo comparativo de descritores para recuperação de imagens por conteúdo na web**. Dissertação (Mestrado em Ciência da Computação) – Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2009.

PHILLIPS, Jeff M. **L7 Distances**. Disponível em: <<http://www.cs.utah.edu/~jeffp/teaching/cs5955/L7-Distances.pdf>>. Acesso em: 25 nov. 2014.

PORKAEW, Kriengkrai; CHAKRABARTI, Kaushik. **Query refinement for multimedia similarity retrieval in MARS**. In: Proceedings of the seventh ACM international conference on Multimedia (Part 1). ACM, 1999. p. 235-238.

RUI, Yong et al. **Relevance feedback: a power tool for interactive content-based image retrieval**. Circuits and Systems for Video Technology, IEEE Transactions on, v. 8, n. 5, p. 644-655, 1998.

SANTOS, Clézio. O uso de desenhos no ensino fundamental: imagens e conceitos. In: PONTUSCHKA, Nídia Nacib; OLIVEIRA, Ariovaldo Umbelino (Org.). **Geografia em perspectiva: ensino e pesquisa**. São Paulo: Contexto, 2002. p. 195-206. Disponível em: <[http://www.qrce.com.br/portal\\_educacional/fundamental2/projeto\\_apoema/pdf/textos\\_complementares/geografia/6\\_ano/apg6\\_texto\\_complementar3.pdf](http://www.qrce.com.br/portal_educacional/fundamental2/projeto_apoema/pdf/textos_complementares/geografia/6_ano/apg6_texto_complementar3.pdf)>. Acesso em: 22 nov. 2015.

SMEULDERS, Arnold WM et al. **Content-based image retrieval at the end of the early years**. Pattern Analysis and Machine Intelligence, IEEE Transactions on, v. 22, n. 12, p. 1349-1380, 2000.

TORRES, Ricardo da S.; FALCÃO, Alexandre X. **Recuperação de Imagens Baseadas em Conteúdo**. In: Workshop de Visão Computacional. 2006.

\_\_\_\_\_. **Recuperação de Imagens Baseadas em Conteúdo**. In: Workshop de Visão Computacional. 2008.