



Henrique Manuel Monteiro Bernardo Corte

Licenciado em Ciências da Engenharia
Eletrotécnica e de Computadores

**Plataforma online de partilha e comparação de
software para análise de imagem da retina**

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e Computadores

Orientador: Professor André Damas Mora, FCT-UNL

Júri:

Presidente: Prof. Doutor José Manuel Matos Ribeiro da Fonseca

Arguentes: Prof. Doutor Tiago Oliveira Machado de Figueiredo
Cardoso

Vogais: Prof. Doutor André Teixeira Bento Damas Mora



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2014

Plataforma online de partilha e comparação de software para análise de imagens da retina

Copyright © Henrique Manuel Monteiro Bernardo Corte, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Com a finalização do meu curso Mestrado Integrado em Engenharia Electrotécnica e Computadores, realizei um trabalho difícil mas gratificante, que seria impossível sem a ajuda de várias pessoas.

Começo por agradecer ao meu coordenador, o Professor Doutor André Damas Mora, pela sua total disponibilidade em ajudar-me a esclarecer as minhas dúvidas ao longo da Tese.

Gostaria de agradecer a todos os colegas de Curso que me ajudaram ao longo deste percurso Académico e na finalização deste. Em especial aos diferentes elementos dos grupos a que pertenci nas diferentes cadeiras do curso. Agradeço também a todos aqueles com que convivi durante os intervalos que me ajudaram a descontraír e a recuperar para voltar de novo ao trabalho.

Por fim gostaria de agradecer aos meus pais e irmãos que me apoiaram sempre neste percurso académico.

Resumo

Nesta Tese foi desenvolvida uma *plataforma online* multiutilizador, que tem como objetivo principal comparar algoritmos de análise de imagens para determinar o seu grau de eficácia. Um exemplo de aplicação é a comparação de algoritmos de análise de imagens da retina para deteção de drusas. A comparação é feita considerando um dos algoritmos como referência padrão e sobre o este são avaliados os restantes.

O funcionamento da plataforma é semelhante à de um fórum, onde é possível a um utilizador criar tópicos publicando imagens e seu descritivo. Após a criação do tópico qualquer utilizador pode visualizar o mesmo, dando a hipótese de comentar ou de acrescentar imagens processadas com os seus próprios algoritmos de análise. Com o aumento de imagens processadas, obtém-se uma base de dados de algoritmos de análise de imagens sobre a qual é possível avaliar o seu grau de eficácia.

A plataforma pretende também criar comunidades onde os utilizadores possam interagir uns com os outros comentando sobre os tópicos, contribuindo assim para o melhoramento dos algoritmos. Deste modo, além de uma base de dados que qualquer utilizador pode usar, obtém-se uma fonte de informação disponibilizada por outros profissionais da área.

Palavras-chave: Processamento de imagem, Plataforma *online*, Eficácia de algoritmos.

Abstract

In this Thesis a multi-user online platform was developed, whose main objective is to compare image analysis algorithms in order to determine their degree of effectiveness. An example of an application is the comparison of retinal image analysis algorithms for the detection of drusen. The comparison is made considering one algorithm as the gold standard and the remaining ones are compared with it.

The platform works similarly to an online forum, where a user can create topics by publishing images and their description. After the topic creation, any user access it, giving him the possibility to comment or add processed images with his own analysis algorithm. With the increase of submitted processed images, a data base of image analysis algorithms, where the degree of effectiveness can be evaluated, is obtained.

The platform promotes also the creation of user communities where the users can interact with each other by commenting on the topics, contributing to the algorithms improvement. This way, not only a data base that any user can use is obtained, but also an information source shared by other experts of the same scientific area is made available.

Key-Words: Image Processing, Online Platform, Algorithms efficiency.

Conteúdo

1	INTRODUÇÃO	1
2	ESTADO DA ARTE	5
2.1	RETINA IMAGE BANK	5
2.2	IMAGE SCIENCE INSTITUTE	6
2.3	BISQUE	9
2.4	PUBLIC IMAGE DATABASES	10
2.5	ESCOLHA PHP VS C# ASP.NET	12
2.6	SILVERLIGHT	13
3	TRABALHO DESENVOLVIDO	15
3.1	ANÁLISE DE REQUISITOS	15
3.2	BASE DE DADOS	20
3.3	PROCESSAMENTO DIGITAL DE IMAGEM	24
3.3.1	<i>Formatos de Imagens</i>	25
3.3.2	<i>Modelos de Cor</i>	26
3.3.3	<i>Máscaras</i>	27
3.3.4	<i>Imagens em Silverlight</i>	28
3.3.5	<i>Tópicos</i>	28
3.4	INSTALAÇÃO (DEPLOY)	30
4	VALIDAÇÃO	32
4.1	HOME PAGE	33
4.2	ABOUT US	34
4.3	CREATE ACCOUNT	35
4.4	LOGIN	36
4.5	CREATE NEW TOPIC	37
4.6	VIEW TOPICS	38
4.7	SELEÇÃO DE UM TÓPICO	39
4.8	ACCOUNT	41
4.9	ANÁLISE DE RESULTADOS	43
5	CONCLUSÕES E TRABALHO FUTURO	51
6	BIBLIOGRAFIA	52

Lista de Figuras

- FIGURA 1-1: *RANGER 7*
- FIGURA 1-2: OBJECTO 3D COM RECURSO A *CAT*
- FIGURA 2-1: *HOME PAGE* DE *IMAGE BANK*
- FIGURA 2-2: PÁGINA *DRIVE BANK* COM IMAGENS E RESULTADOS
- FIGURA 2-3: VISUALIZAÇÃO DE UM TÓPICO DE RETINA
- FIGURA 2-4: PÁGINA PRINCIPAL DE *BISQUE* COM *LOGIN* JÁ FEITO
- FIGURA 2-5: *NUCLEAR FILTER PAGE*
- FIGURA 2-6: *HOME PAGE* DE *IMAGE DATABASES*
- FIGURA 2-7: LISTA DE EQUIPAS E A RESPECTIVA DATA E RESULTADO OBTIDOS
- FIGURA 2-8: FERRAMENTA *FIDDLER* E A VISUALIZAÇÃO DA COMUNICAÇÃO
- FIGURA 3-1: *PLATAFORMA ONLINE* E INTERAÇÃO DOS VÁRIOS COMPONENTS. (ARQUITETURA DA PLATAFORMA)
- FIGURA 3-2: EXEMPLO DE UM *DATA SET*
- FIGURA 3-3: EXEMPLO DE UM *DATA SET PROCESSADO*
- FIGURA 3-4: EXEMPLO DE *MÁSCARAS* DO *DATA SET*
- FIGURA 3-5: LÓGICA DO PROCESSAMENTO DUMA IMAGEM
- FIGURA 3-6: *MÁSCARAS* DO *DATA SET*
- FIGURA 3-7: DIAGRAMA DE ENTIDADES E RELAÇÕES DA BASE DE DADOS
- FIGURA 3-8: CONJUNTO DE TUPLOS DE UM *DATA SET*
- FIGURA 3-9: CONJUNTO DE TUPLOS DE UM *DATA SET PROCESSADO*
- FIGURA 3-10: IMAGENS JPEG
- FIGURA 3-11: DA ESQUERDA PARA A DIREITA 1BIT/PIXEL, 2BIT/PIXEL, 24 BIT/PIXEL.
- FIGURA 3-12: CUBO DO MODELO DE COR RGB
- FIGURA 3-13: A PRETO SÃO OS *PIXÉIS* DA *MÁSCARA* A INDICAR A ÁREA DE INTERESSE
- FIGURA 3-14: OS *PIXÉIS* A VERDE SÃO OS *PIXÉIS* DETETADOS NUMA IMAGEM IDEAL
- FIGURA 4-1: BARRA DE NAVEGAÇÃO
- FIGURA 4-2: *FRAME*
- FIGURA 4-3: HOME PAGE
- FIGURA 4-4: AVISOS DO ADMINISTRADOR
- FIGURA 4-5: *ABOUT US*
- FIGURA 4-6: *CREATE ACCOUNT*
- FIGURA 4-7: *USER LOGIN*
- FIGURA 4-8: *CREATE NEW TOPIC*
- FIGURA 4-9: VIEW TOPICS
- FIGURA 4-10: TOPIC SELECTED
- FIGURA 4-11: *DATA SET*, *MÁSCARA* E IMAGENS PROCESSADAS
- FIGURA 4-12: RESULTADOS DA COMPARAÇÃO
- FIGURA 4-13: COMENTÁRIOS DE UTILIZADORES
- FIGURA 4-14: MENU DE UTILIZADOR

FIGURA 4-15: DADOS DA CONTA DE UM UTILIZADOR

FIGURA 4-16: MY TOPICS (TÓPICOS CRIADO PELO UTILIZADOR QUE FEZ LOGIN)

FIGURA 4-17: TODAS IMAGENS QUE PERTECEM AO *DATASET* CRIADO.

FIGURA 4-18: LISTA DOS NOMES DAS 22 IMAGENS DO *DATASET*, *DATASET PROCESSADO* E *MÁSCARAS*.

FIGURA 4-19 : PRIMEIRA IMAGEM DO *DATASET*, A *MÁSCARA* E IMAGENS DOS *DATASET PROCESSADOS*.

FIGURA 4-20: TABELA DE RESULTADOS DA COMPARAÇÃO DAS IMAGENS.

FIGURA 4-21: *ZOOM* VISÍVEL DAS QUATRO PRIMEIRAS IMAGENS DOS *DATASET PROCESSADO* ESCOLHIDOS.

FIGURA 4-22: IMAGEM “*TRIAL 001 (19)*”, A *MÁSCARA* E AS IMAGENS DOS *DATASET PROCESSADOS* ASSOCIADAS, SENDO “*ALGORITHM B*” O *DATASET PROCESSADO* CONSIDERADO IDEAL.

FIGURA 4-23: RESULTADOS DA IMAGEM “*TRIAL 001 (19)*” E O *DATASET PROCESSADO* “*ALGORITHM B*” ESCOLHIDO COMO CASO IDEAL.

FIGURA 4-24: IMAGEM “*TRIAL 001 (19)*”, A *MÁSCARA* E AS IMAGENS DOS *DATASET PROCESSADOS* ASSOCIADAS, SENDO “*ALGORITHM F*” O *DATASET PROCESSADO* CONSIDERADO IDEAL.

FIGURA 4-25: RESULTADOS DA IMAGEM “*TRIAL 001 (19)*” E O *DATASET PROCESSADO* “*ALGORITHM F*” ESCOLHIDO COMO CASO IDEAL.

Lista de Tabelas

TABELA 3-1: REQUISITOS FUNCIONAIS	ERRO! MARCADOR NÃO DEFINIDO.
TABELA 3-2: FORMATO DE IMAGENS PERMITIDO.....	17
TABELA 3-3: TABELA DEPOSITANTE	21
TABELA 3-4: TABELA CONTA	21
TABELA 3-5: TABELA DEPOSITANTE NORMALIZADA	21
TABELA 3-6: VARIÁVEIS E FÓRMULAS NA COMPARAÇÃO ENTRE DUAS IMAGENS.....	29
TABELA 4-1: COEFICIENTE DE INTREPRETAÇÃO COHEN'S KAPPA.....	46

Acrónimos

- WCF** - *Windows Communication Foundation*
- BD** - *Base de Datos*
- IDE** - *Integrated Development Environment*
- XAML** - *Extensible Application Markup Language*
- DBMS** - *Database Management System*
- SQL** - *Structured Query Language*
- PHP** - *Hypertext Preprocessor*
- DLL** - *Dynamic Link Library*
- RGB** - *Red Green Blue*
- RIA** - *Rich Internet Application*
- IIS** - *Internet Information Services*
- GAC** - *Global Assembly Cache*
- JPEG** - *Joint Photographic Experts Group*
- GIF** - *Graphics Interchange Format*
- BMP** - *Bitmap Image File*
- TIFF** - *Tagged Image File Format*

1 Introdução

O aparecimento do processamento de imagens digitais provém do início de 1960 tendo como impulsionador principal o programa espacial. No *JET PROPULSION LABORATORY* começaram a aplicar técnicas para melhoramento de imagens, provenientes da lua, transmitidas pela sonda espacial *RANGER 7* (figura 1-1). O melhoramento ou correção na imagem era a remoção de vários tipos de distorções que foram introduzidos pela câmara na captação das imagens.

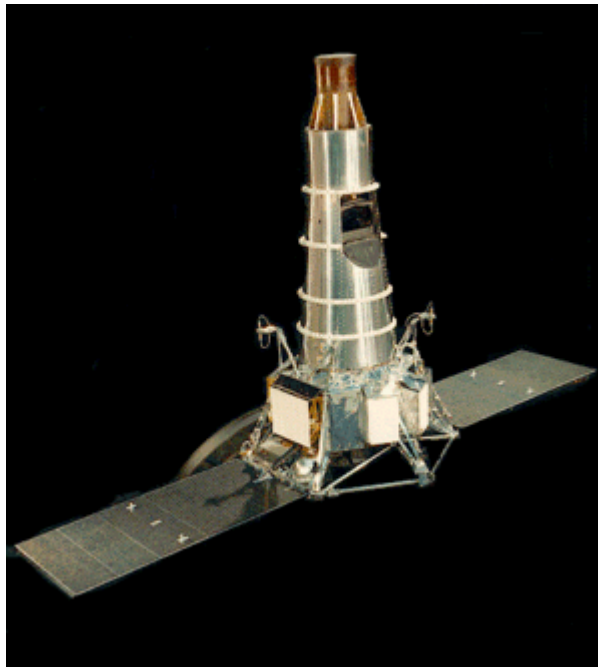


Figura 1-1: *RANGER 7*

No início de 1970 as aplicações de processamento de imagens digitais aumentaram, passando a ser usado em imagens médicas, em imagens de astronomia, na observação de recursos terrestres em áreas remotas, entre outras.

A área da Medicina, especialmente a área de Diagnóstico, teve uma grande evolução com o aparecimento de exames por imagem. Estes exames permitem visualizar partes do corpo humano sem a necessidade de uma operação invasiva ou minimizando esta. Os exames mais conhecidos e realizados são o *raio-X*, *ultrassom* e o *CAT* [1].

O aparecimento de *CAT* (*computerized axial tomography*) ou *CT* (*computerized tomography*) para abreviar. Foi um dos mais importantes avanços no processamento de imagens em diagnósticos médicos. Funciona com o recurso de *Raio-X* e algoritmos para recriar a partir dos dados obtidos uma imagem que é uma fatia do objeto, ao juntar todas as fatias obtidas obtém-se o objeto em 3D (figura 1-2).

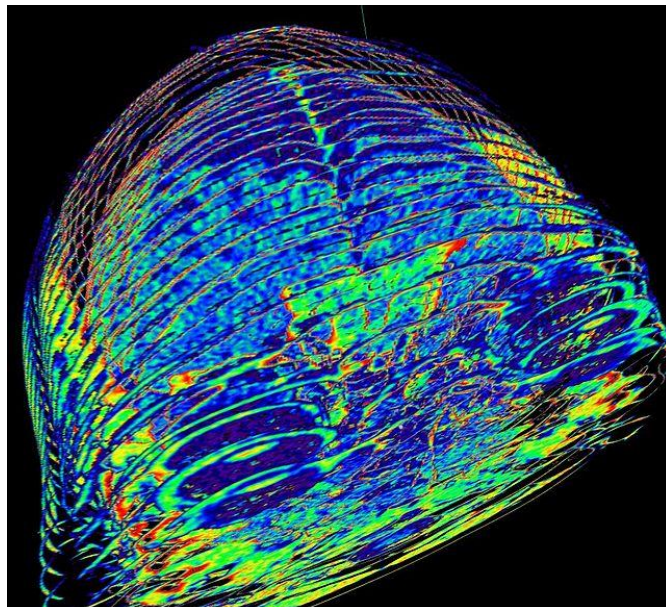


Figura 1-2: Objeto 3D com recurso a CAT

Com uma maior rapidez no diagnóstico, é possível combater a doença mais eficazmente de modo a não agravar ou prevenindo o seu aparecimento. Pois diagnosticar uma doença através de sintomas já existentes faz com que muitas vezes seja tarde demais para uma recuperação.

Através desta análise de imagens e diagnósticos mais rápidos, o custo do tratamento é reduzido significativamente para além dos benefícios sociais, sendo os testes de triagem mais baratos e eficientes. [2]

Para esse efeito, hoje em dia existem bastantes estudos sobre análises de imagens na área da medicina. Muitos desses estudos encontram-se em plataformas *online* [3] [4], que comparam algoritmos de processamento digital retornando dados relevantes para interpretação de qual será o melhor algoritmo. Esses algoritmos poderão ir desde o melhoramento da imagem para facilitar a sua visualização, até à deteção de anomalias que se encontram na imagem para diagnóstico precoce de doenças.

Com o contínuo aparecimento destas plataformas *online* será possível obter cada vez mais um conjunto de bases de dados de algoritmos de análise de imagens médicas bem como a diversificação nas áreas em que essas análises ocorrem. Sendo filtrados os melhores algoritmos é possível aumentar a taxa de diagnóstico correto investindo nos algoritmos que tenham potencial.

Num futuro não muito distante, o diagnóstico poderá ser feito automaticamente através do uso de algoritmos que interpretem os sintomas dos pacientes. Um grande contributo serão as plataformas *online* através do armazenamento desses algoritmos em base de dados e a avaliação do seu grau de eficácia. O grau de eficácia depende da deteção correta de anomalias, um exemplo seria a deteção correta de drusas na retina de um paciente [5].

Ir-se-á dividir esta tese em cinco capítulos: Introdução, Estado da Arte, Trabalho desenvolvido, Validação, Conclusões e Trabalho futuro.

O primeiro capítulo de Introdução constituiu uma breve introdução ao aparecimento do processamento de imagens, tendo-se posteriormente referido a sua importância na área de medicina e as suas vantagens. Por fim, foi abordado o aparecimento de plataformas *online* ligadas à área da medicina e de processamento digital.

O segundo capítulo de Estado da Arte será repartido em duas componentes. Na primeira descrever-se-ão duas *plataformas online*, o seu funcionamento geral e algumas das suas limitações. A segunda componente focar-se-á nas tecnologias usadas para a construção da *plataforma online* desenvolvida nesta tese e na argumentação da escolha dessa tecnologia em detrimento de outras.

No capítulo de Trabalho desenvolvido será apresentada a análise de requisitos, a base de dados de suporte à *plataforma online* e posteriormente um esquema geral da arquitetura da plataforma e uma breve descrição do seu funcionamento.

No quarto capítulo Validação será feita uma descrição sucinta da *plataforma online* e de todas as operações. Serão referidas também algumas regras que o utilizador deve seguir e o funcionamento da *plataforma online*.

Por último no capítulo das Conclusões e Trabalho futuro serão descritas as vantagens inerentes ao uso da plataforma e serão apresentados alguns melhoramentos possíveis de se efetuarem à plataforma.

2 Estado da Arte

Com o aparecimento de um maior número de *plataformas online* focadas na área de medicina é necessário haver uma proporcional inovação e melhoramento em relação às plataformas já existentes.

Para tal deve fazer-se um estudo prévio das *plataformas* existentes, visitando as mesmas e vendo os aspetos positivos que se possam adotar na plataforma que se vai desenvolver. De seguida, são apresentadas as *plataformas online* estudadas.

2.1 Retina Image Bank

Retina Image Bank é um projeto da *American Society Retina Specialists*, lançado em agosto de 2012 (figura 2-1). É uma plataforma de armazenamento de imagens da retina cujo objetivo é cobrir a área de tratamento, educacional e anatómico [6].

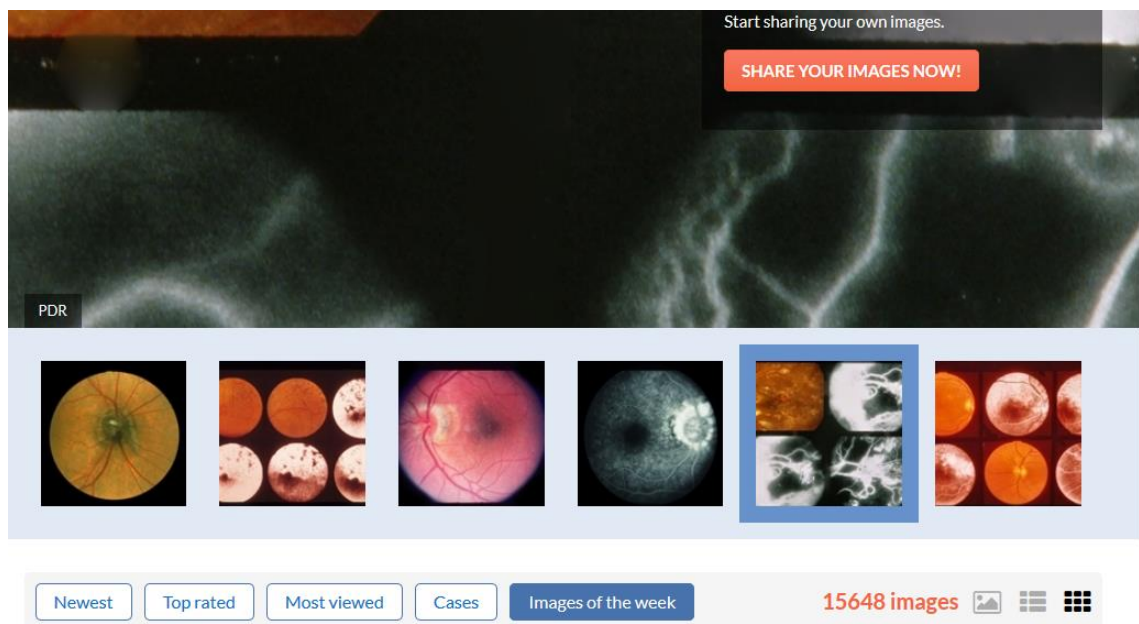


Figura 2-1: Home page de Image Bank

Em caso de dúvida, poder-se-á esclarecer visualizando o guia disponibilizado. Para fazer *upload* e *download* de imagens é conveniente uma leitura prévia do guia. Para o *upload* de imagens é dito que qualquer membro de ASRS é livre de fazer *upload*, mas para quem não seja membro tem que contactar primeiro os escritórios da ASRS para o

registo de dados. Posteriormente para fazer *uploads* das imagens é indicado o tamanho máximo de 50MB por imagem e os formatos de *JPEG* e *PNG*. Após esta etapa, basta pressionar no botão *Upload*, escolher *Upload files*, selecionar as imagens a transferir e efetuar *start upload*.

No caso de *download*, cada utilizador inscrito que não seja membro de *ASRS* pode efetuar o *download* de 3 imagens por *login*, caso seja membro de *ASRS* ou filiado à *ASRS* passa a ser 10 imagens. Para além do limite de imagens por *login* também tem que se incluir os direitos de autor ao usar as imagens, tal como usar as imagens para fins informativos e não comerciais, nem disponibilizar para o público em geral. Por fim não fazer nenhuma alteração na imagem ou no seu conteúdo escrito. Para fazer *download* basta carregar em qualquer imagem disponibilizada em *Home* e carregar no botão de *download* que indica o tamanho da imagem.

2.2 Image Science Institute

O *Image Sciences Institute* é um instituto que pertence à Universidade de Utrecht. Cujo principal objetivo é fazer investigação relacionada com imagens médicas.

Este instituto tem uma *plataforma online*, na qual é possível visualizar os diferentes casos de estudos. Esses casos referem-se a imagens médicas que são processadas por vários algoritmos diferentes e donde são retirados dados das respetivas comparações [3].

As imagens médicas encontram-se armazenadas em várias bases de dados que a plataforma contém. Algumas das base de dados são: *AMIDA13*, *DRIVE DATABASE* e *EMPIRE10* que se encontram disponibilizados na opção *DataBases* que se encontra no menu principal da plataforma para serem visualizados

No caso do *DRIVE database* é um conjunto de imagens da retina sobre as quais são aplicadas técnicas de processamento de imagem para deteção de vasos sanguíneos. Para se visualizar todo o conjunto de imagens tem que se selecionar todas as *checkboxes* (*input, gold standard, human observer, Chaudhuri etc.*) Figura 2-2.

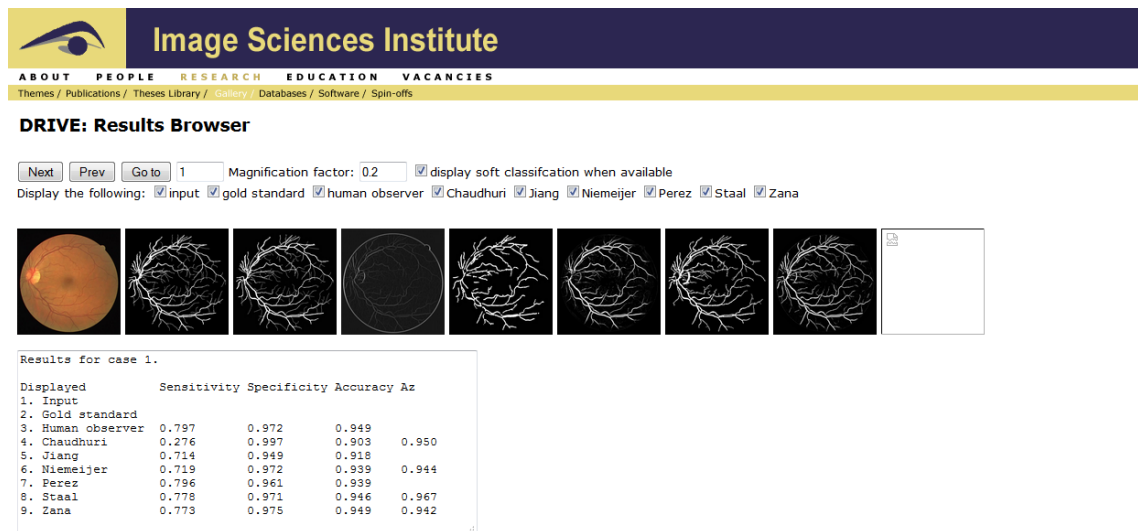
Sendo *input* a imagem original sem alterações nenhuma e *Golden Standard* a imagem ideal com a qual as restantes vão ser comparadas. Após a seleção das *checkboxes* além do aparecimento das respetivas imagens também são disponibilizados dados estatísticos da comparação entre imagens, nomeadamente: *Sensitivity, Specificity, Accuracy, Az* [7].

Sensitivity é a comparação direta entre a imagem *Golden Standard* e a respetiva a imagem que se encontra a ser comparada *pixel a pixel* em relação ao total de *pixéis* detetados na *Golden Standard*.

Specificity é relação que indica que o número de *pixéis* incorretamente detetados (Falsos Positivos) na imagem que se encontra a ser comparada com a imagem *Golden Standard* é um valor significativo ou não.

Accuracy é a soma do total de *pixéis* detetados corretamente de vasos retinianos e os *pixéis* que não são de vasos retinianos dividido pelo número total de *pixéis* de interesse da retina.

Az é a relação entre número de verdadeiros positivos em relação ao número de falsos positivos.



Displayed	Sensitivity	Specificity	Accuracy	Az
1. Input				
2. Gold standard				
3. Human observer	0.797	0.972	0.949	
4. Chaudhuri	0.276	0.997	0.903	0.950
5. Jiang	0.714	0.849	0.918	
6. Niemeijer	0.719	0.972	0.939	0.944
7. Perez	0.796	0.961	0.939	
8. Staal	0.778	0.971	0.946	0.967
9. Zana	0.773	0.975	0.949	0.942

Figura 2-2: Página *Drive bank* com imagens e resultados

Cada conjunto de imagens disponibilizadas é estático e já se encontra na Plataforma. Não sendo possível acrescentar mais pelo utilizador, mas pode fazer-se *download* desse conjunto de imagens.

Com esta Plataforma os investigadores têm mais uma ferramenta para complementar a sua investigação.

Esta plataforma mais especificamente a página do *DRIVE database* foi usada como referencia para o desenvolvimento da plataforma da tese na visualização de um tópico como o que se encontra na figura 2-3.

Main Page About us Create New Topic View Topics www Log Out

Masks upload included

Summary
Masks to get one of the values needed and simplify the process

Add as Favorite Topic
Add

+ - a
28/6/2014

Download DataSet Upload Processed DataSet

Download DataSet Upload Processed DataSet

Next Go to

AlgorithmA AlgorithmB AlgorithmD

Figura 2-3: Visualização de um tópico de retina

2.3 Bisque

Bisque que se encontra na figura 2-4 é um repositório *online* de imagens biológicas que pertence à Universidade da Califórnia em Santa Bárbara [4], tendo um conjunto de técnicas ou algoritmos incorporados no seu sistema para análise de imagens [8]. Foi desenvolvido pelo Centro para bioinformática que é um grupo de investigação e desenvolvimento de múltiplas áreas de biologia, engenharia e multimédia, com o propósito de aumentar o conhecimento no funcionamento de processos biológicos complexos a nível celular.

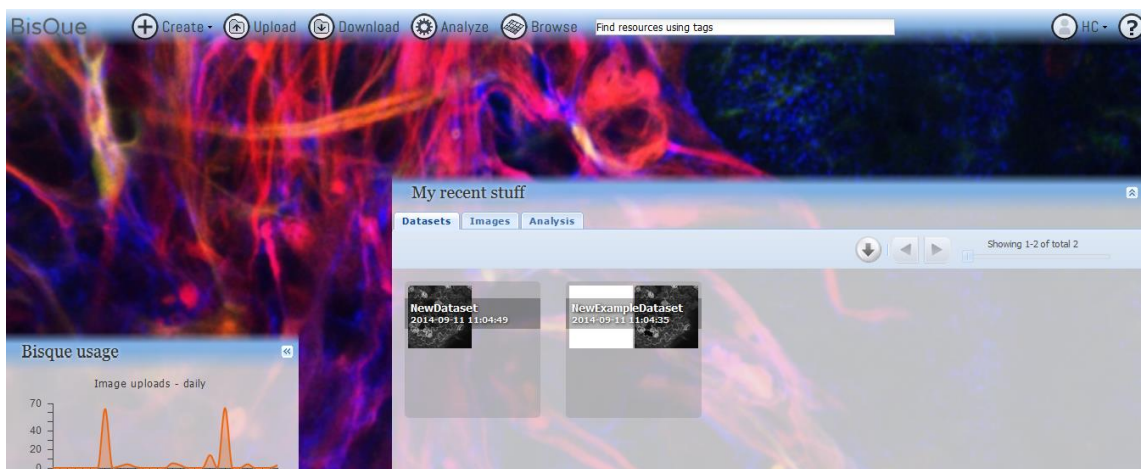


Figura 2-4: Página principal de *Bisque* com *login* já feito

Para se poder ter acesso à maior parte das funcionalidades do sistema é necessário criar uma conta e fazer *login*, sendo possível fazer *upload* de um conjunto de imagens e criar um *DataSet*.

Para se fazer uma análise de uma imagem ou conjunto de imagens basta carregar em *Analyze* e escolher um dos processos disponibilizados. Para cada processo que se escolha está associado um conjunto de parâmetros necessários para o seu funcionamento, como se encontra na figura 2-5, sendo disponibilizado um pequeno guia para ensinar como usar ou para que serve o processo.

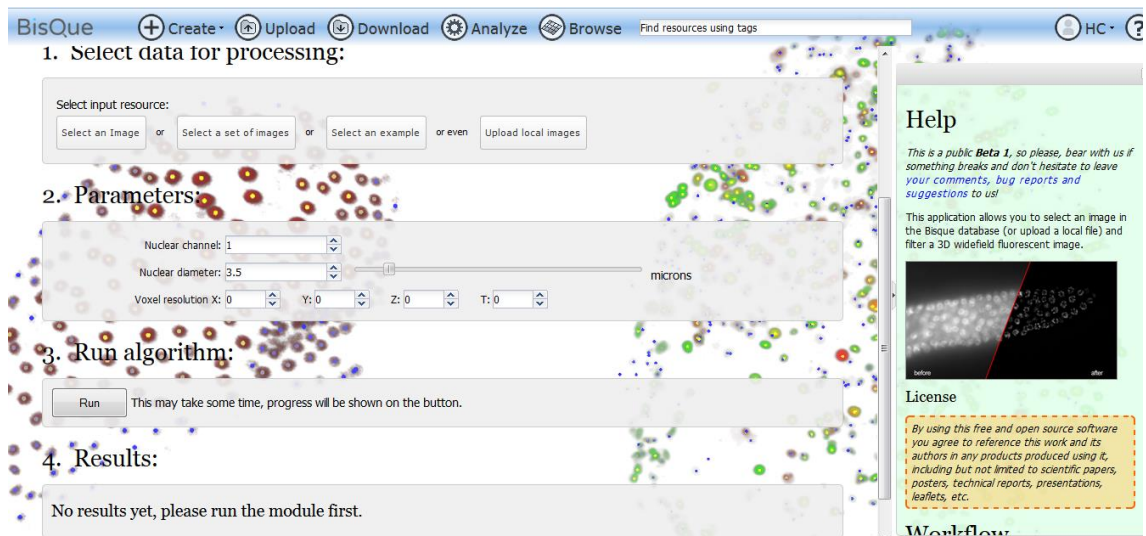


Figura 2-5: Nuclear Filter Page

O sistema contém muitas funcionalidades, no entanto apresenta algumas desvantagens nomeadamente que o conjunto de processos de análises de imagens é fixo, sendo necessário ter conhecimento na área de modo a preencher os parâmetros e que não se pode adicionar novos métodos de processamento de imagens.

2.4 Public Image Databases

O *Public images Databases* é uma *Web page* da Universidade *Cornell* desenvolvida pelo grupo de análise de imagens e visão [9], que tem vários *links* para bases de dados públicas de imagens médicas e contém uma área para análise de imagens médicas tendo alguns casos de estudos que se encontram na figura 2-6.

Medical Image Analysis Challenges

Challenges, sometimes termed grand challenges provide data sets for the purpose of comparison of different analysis methods. While these challenges do provide a resource for image data they may often incur more restrictive conditions on how the data may be used.

Limited Data Access unless otherwise noted, see individual challenge for details

Computed Tomography (CT)

- EXACT09. Challenge for the automated extraction of airways from CT data. [EXACT09 Homepage](#)
- ANODE09 Study. Challenge for the CAD detection of lung lesions from whole-lung CT data. [ANODE09 Homepage](#)
- BIOCHANGE 2008 PILOT. Challenge for the evaluation of change measurement algorithm. [BIOCHANGE 2008 Pilot Homepage](#) *Note: Uses publicly accessible NCIA-RIDER and FDA phantom data.
- VOLCANO'09 Nodule Change Challenge the evaluation of size change in pulmonary nodules.
- Coronary Artery Algorithm Evaluation Framework. Challenge based on the extraction of coronary artery centerlines from CTA data. [Home Page](#)
- Liver Tumor Segmentation 08. Challenge based on the segmentation of liver lesions from contrast enhanced CT. [Challenge Homepage](#)

Figura 2-6: Home page de Image Databases

Estes casos de estudos estão limitados aos *DataSet* predefinidos não se podendo inserir nenhum *DataSet* novo. Para se poder fazer *download* do *DataSet* em estudo tem que se inscrever uma equipa e após processar as imagens desse *DataSet* com o algoritmo desenvolvido ou método desenvolvido pode fazer-se o *upload* para o caso em estudo de modo a ser avaliado, sendo disponibilizada uma lista de equipas e respetiva pontuação que obtiveram, como é mostrado na figura 2-7.

Results

Below the submitted results so far are listed. Click on the score of a method to see detailed results.

Methods that have checkboxes in front of them can be combined. Check a number of boxes and press Combine. The combination method that is used is described in [this report](#).

Team	Date	Score	Description
<input type="checkbox"/> ISICAD	December 20, 2008	0.632	System E in this paper .
<input type="checkbox"/> FlyerScan	September 25, 2009	0.552	System description .
<input type="checkbox"/> MIG Group	August 3, 2010	0.310	Not yet publicly available.
<input type="checkbox"/> Pisa team	January 26, 2009	0.293	System D in this paper .
<input type="checkbox"/> RG-MAGIC-CAD	May 21, 2009	0.291	System B in this paper .
<input type="checkbox"/> MIG Group	August 3, 2010	0.289	Not yet publicly available.
<input type="checkbox"/> RG-MAGIC-CAD	January 20, 2009	0.266	System description .
<input type="checkbox"/> MAGIC-5-ANTS	January 23, 2009	0.254	System C in this paper .
<input type="checkbox"/> Philips Lung Nodule CAD	May 29, 2009	0.231	System F in this paper .
<input type="checkbox"/> FujitaLab	August 26, 2008	0.212	System A in this paper .
<input type="checkbox"/> CMP Prague	December 24, 2008	0.065	System description .
<input type="checkbox"/> IMPS	July 30, 2008	0.003	System description .

Figura 2-7: Lista de equipas e a respetiva data e resultado obtidos

Após a visualização de algumas *plataformas online* de partilha de imagens e algoritmos de processamento serão vistas as tecnologias usadas na construção e desenvolvimento da plataforma *SIDAC*.

2.5 Escolha PHP vs C# ASP.NET

Para o desenvolvimento da plataforma *online* foi escolhida a tecnologia *Silverlight*. A escolha do *Silverlight* e *C# ASP.NET*, em relação a *PHP* foi por um lado a vontade de aprender uma nova tecnologia e por outro adquirir conhecimentos na *framework .NET* e aprofundar o conhecimento em *C#*. Ao ter um maior à vontade em *C#* na com-

ponente de manipulação de imagens na plataforma a ser desenvolvida. *PHP* é uma linguagem criada e usada especificamente para a criação de *Websites* enquanto *C#* é uma linguagem que tanto pode ser usada na criação de uma aplicação de *desktop* como para criação de um *Website* ou *plataforma online*. Outra vantagem do *Silverlight* é que é uma tecnologia que corre no lado do cliente ao contrário de *PHP* que corre no lado do servidor. Desta forma não é necessário fazer *reload* da página total mas só partes específicas através de *frames*. [10] [11].

2.6 Silverlight

Silverlight é uma *application framework* que tem suporte para várias plataformas e navegadores, sendo utilizada para criar *aplicações ricas para a Internet (RIA)*. As principais linguagens de programação com que trabalha são: *Extensible Application Markup Language (XAML)* e *C sharp (C#)* [12].

XAML é uma linguagem de interface da *Microsoft*, sendo uma linguagem de interface diz-se que corre à frente, pois quaisquer alterações são diretamente visíveis. Enquanto *C#* corre atrás, com isto quer dizer-se que não serão tão visíveis as suas alterações, pois será mais focado na manipulação de dados. Ter-se dito que não eram tão visíveis as alterações quanto seria esperado, isso será porque com a manipulação de dados ir-se-á também influenciar o *Layout* embora em menor grau. Também é possível desempenhar o papel de *XAML* de definir ou atualizar o *Layout* caso seja pretendido, embora o nível de complexidade do código seja maior sendo por isso evitado, exceto em casos específicos em que se quer o *Layout* mais dinâmico [13].

Silverlight é uma tecnologia que corre no lado do cliente e a sua comunicação com o lado do servidor é feita através de *webservices* mais especificamente *Windows Communication Foundation (WCF)* [14]. Essa comunicação é feita assincronamente, sendo possível assim tornar impercetível ao utilizador que essa comunicação esteja a ocorrer, pois não será interrompido o funcionamento normal do programa [12] [15].

Como já foi referido o *Silverlight* é uma *framework* desenvolvida pela *Microsoft* e encontra-se integrado no *Visual Studio IDE*, que foi usado para o desenvolvimento da plataforma *online*. Tem a vantagem de se poder fazer *debug* exceto no caso da comunicação com o *Web Service*, que embora possa retornar alguns tipos de erros se forem relacionados com a base de dados, se forem erros de comunicação tem que se recorrer a uma ferramenta denominada *Fiddler* [16]. Recorre-se ao *Fiddler* para escutar as comunicações entre o *pc* e o servidor sendo possível ver a mensagem de erro ou de sucesso que é retornada através da opção *raw* do *fiddler*. A mensagem recebida na Figura 2.8 é

um exemplo da plataforma a comunicar com um *Web Service*, neste caso a mensagem recebida é de 200 e *ok* que está no formato *http 1.1*, neste não ocorreu nenhum erro.

De modo a poder fazer-se *debug* usando *Silverlight* quando se usa *Firefox* como *browser* tem de se escrever “*about:config*” no *Browser* e mudar a propriedade “*dom.ipc.plugins.enabled*” de *true* para *false*, caso contrário é ignorado qualquer *break point* que venha a encontrar. Há *patches online* que corrigem essa limitação.

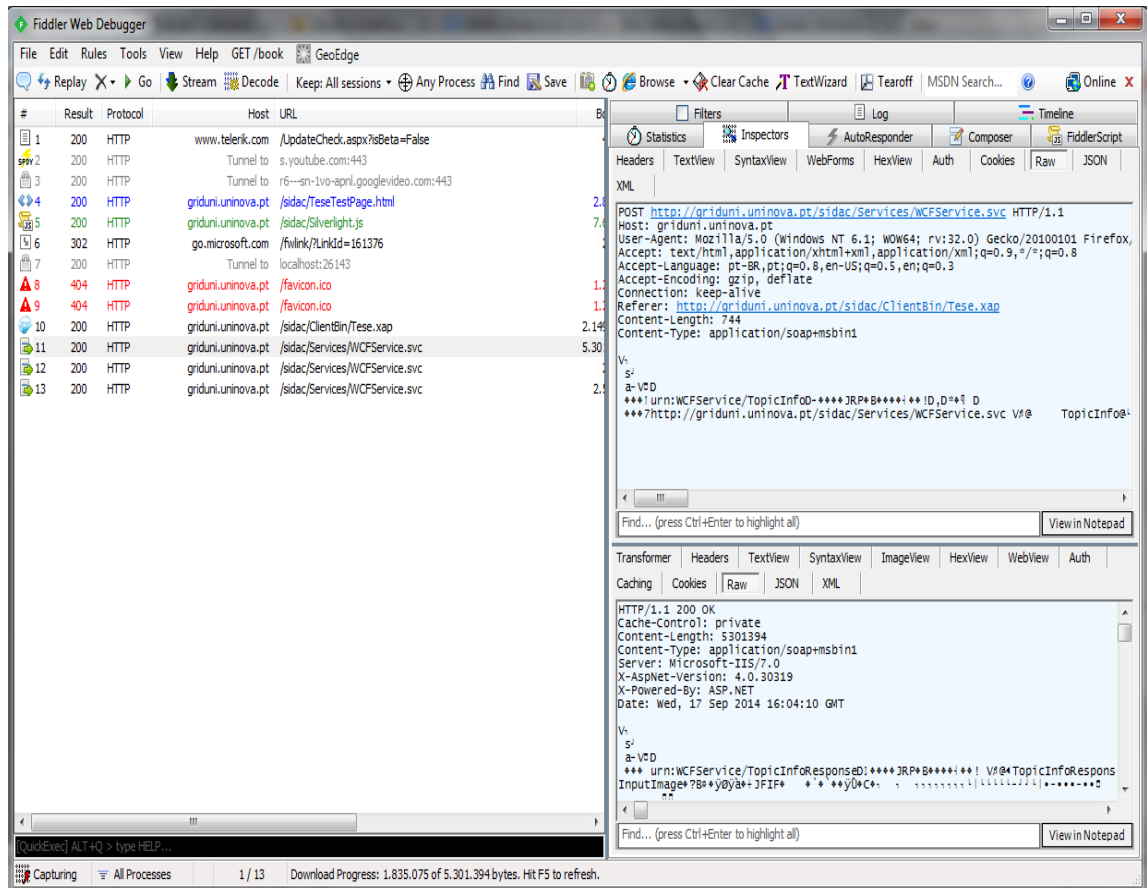


Figura 2-8: Ferramenta *fiddler* e a visualização da comunicação

3 Trabalho desenvolvido

Para desenvolver a *plataforma online* começou-se por estudar o que se pretende que seja capaz de fazer e como funcionará, com essa finalidade fez-se uma análise dos requisitos. Após a análise de requisitos ir-se-á apresentar a base de dados e o processamento digital da imagem neste capítulo.

3.1 Análise de Requisitos

Para se fazer uma análise de requisitos começa-se por criar um conjunto de questões relevantes e a responder a essas questões, que neste caso foram as seguintes:

Onde irá ser usado o Sistema?

A Plataforma será usada na área de medicina mais especificamente em comparação de imagens da retina.

Como irá o Sistema efetuar o pretendido?

O Sistema irá efetuar o pretendido através do armazenamento de dados numa base de dados e ficheiros comprimidos numa pasta no servidor. Sendo os nomes dos ficheiros e dados relevantes para obtenção correta dos ficheiros que se encontram na base de dados.

Quais são os parâmetros críticos do Sistema para o seu funcionamento?

Os parâmetros críticos da plataforma são: armazenamento de grandes quantidades de imagens, capacidade de associação de diferentes conjuntos de imagens entre si, capacidade de comparação de imagens para obtenção de dados.

Qual o grau de eficácia que o sistema terá no seu funcionamento?

A plataforma terá que ser rápida no processamento de dados e imagens e ter capacidade de armazenar grandes conjuntos de imagens.

Por quanto tempo o sistema será usado (pelo cliente)?

Não existe tempo limite definido.

Em que ambientes se espera que o sistema irá funcionar?

O ambiente no caso desta plataforma será o Sistema Operativo e Navegador. O Sistema Operativo será *Windows* e o Navegador *Firefox*. Mas se for possível um maior número de Sistemas Operativos e Navegadores a suportarem a plataforma melhor, na medida em que maior número de pessoas poderá aceder à plataforma.

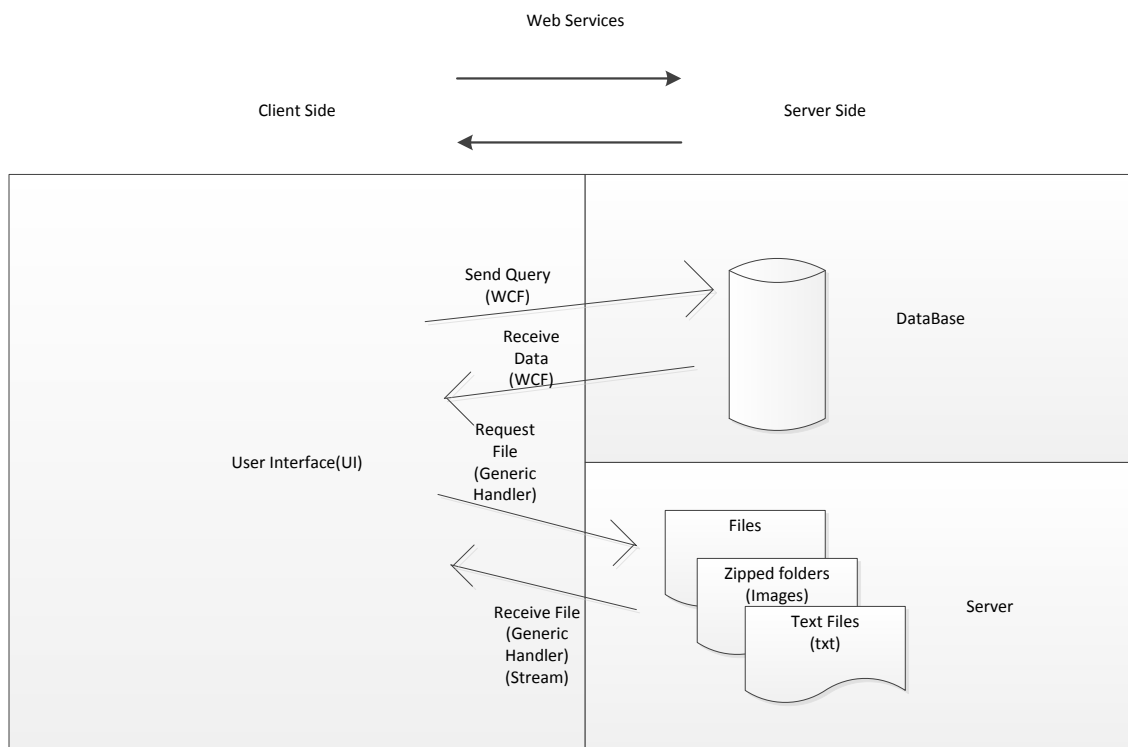


Figura 3-1: Plataforma online e interação dos vários componentes. (arquitetura da plataforma)

A Plataforma *online* divide-se em 3 componentes *user interface*, *database* e *server* que se encontram no diagrama da figura 3-1. A componente *UI* é a lógica da interação do utilizador com a plataforma *online*, proveniente dessa interação é necessário muitas vezes atualizar a *frame* principal para visualizar uma nova página. Para atualizar a *frame* do *Layout* da Plataforma é necessário recolher dados relevantes que se encontram armazenados na base de dados.

Para além dos dados da base de dados é necessário em alguns casos obterem-se ficheiros, que não se encontram na base de dados, mas sim no servidor. Esses ficheiros ou são imagens no formato definido na tabela 4.1 que se encontram comprimidos em pastas no formato *zip* ou são ficheiros cujo formato é definido pelo utilizador que contém o código do algoritmo. Para se obterem os ficheiros corretamente, é necessário saber o nome do ficheiro que se encontra armazenado na base de dados.

Para o funcionamento da plataforma há uma “*constante*” comunicação entre o programa que se encontra no lado do cliente com o servidor e com a base de dados em resposta à interação com os utilizadores. Como já foi referido a comunicação é assíncrona para passar despercebida pelos utilizadores. Essa comunicação é feita através do

Web Service, mais concretamente do *WCF* que é capaz de comunicar informação mais complexa no formato de classes. A comunicação com a base de dados é feita através do envio de *queries* que são processados pela base de dados, retornando os dados sob a forma de tabelas.

(De referir que se irão usar muito os termos *DataSet*, *DataSet Processados* e *Máscaras* e por esse motivo ir-se-á fazer uma breve explicação sobre o que são. O seu uso é posteriormente explicado no contexto apropriado)

DataSet é um conjunto de imagens sobre qualquer tema no formato definido na Tabela 3-2.

Tabela 3-1: Formato de imagens permitido

Portable Network Graphics	Joint Photographic Experts Group	Bitmap	Tagged Image File Format
png	jpg/jpeg	bmp	tif/tiff

Um exemplo de um *Dataset* é o conjunto de imagens da retina visível na Figura 3-2.

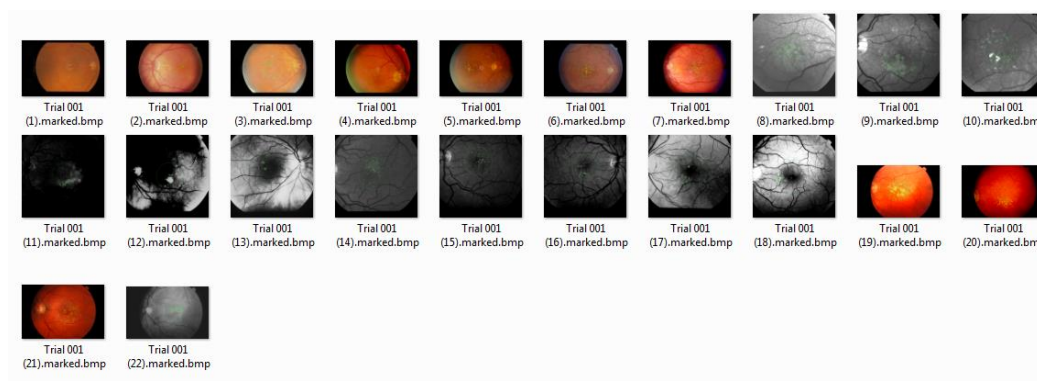


Figura 3-2: Exemplo de um *DataSet*

DataSet Processado é um conjunto de imagens que foram obtidas através do processamento das imagens do *DataSet*. Este processamento é realizado pelo utilizador com o uso do seu próprio *software* e submete as imagens posteriormente. A Figura 3-3 mostra um *DataSet Processado* obtido do *DataSet* da Figura 3-2.

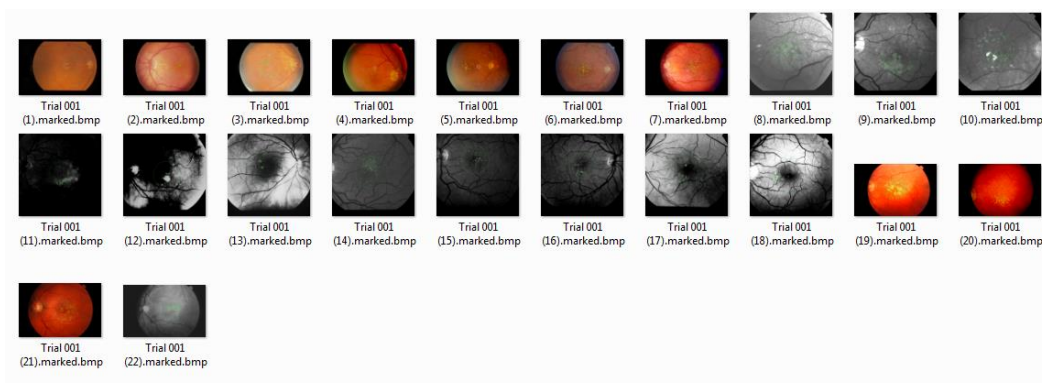


Figura 3-3: Exemplo de um *DataSet* Processado

As *Máscaras* são um conjunto de imagens que representam áreas de interesse de uma imagem e é a zona que irá ser processada. Na figura 3-4 é apresentado o conjunto de *máscaras* do *DataSet* da figura 3-2.

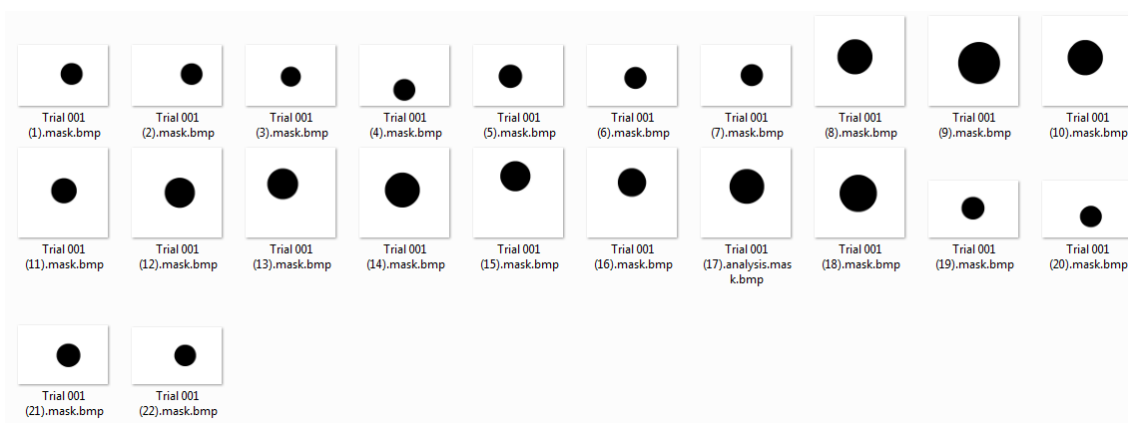


Figura 3-4: Exemplo de *Máscaras* do *DataSet*

Quando se aplica uma *máscara* à imagem original são filtrados todos os dados fora da máscara obtendo-se marcações apenas na área de interesse. O resultado do processamento encontra-se todo contido nessa área, sendo visível a verde o perímetro definido pela *máscara* e as áreas detetadas ou *drusas* a verde dentro dessa circunferência como se pode ver na figura 3-5.

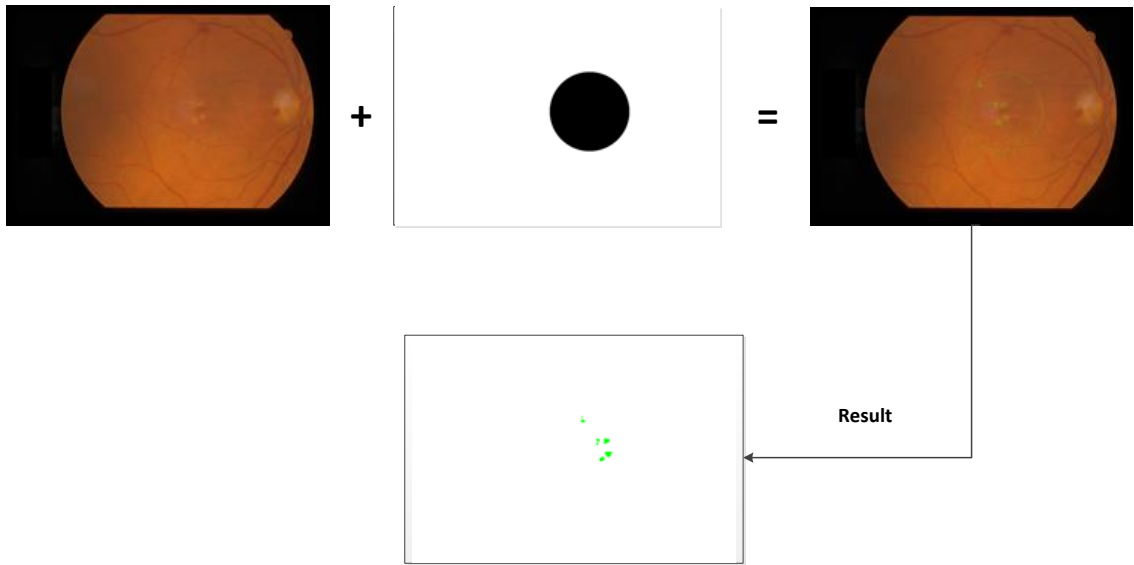


Figura 3-5: Lógica do Processamento numa imagem

A um *DataSet* está associado um ou mais *DataSet Processados* e um conjunto de *Máscaras* como se encontra na figure 3-6. O *DataSet* e as *Máscaras* são definidos por um utilizador ao criar um novo tópico. Os *DataSet Processados* por outro lado são inseridos por outros utilizadores.

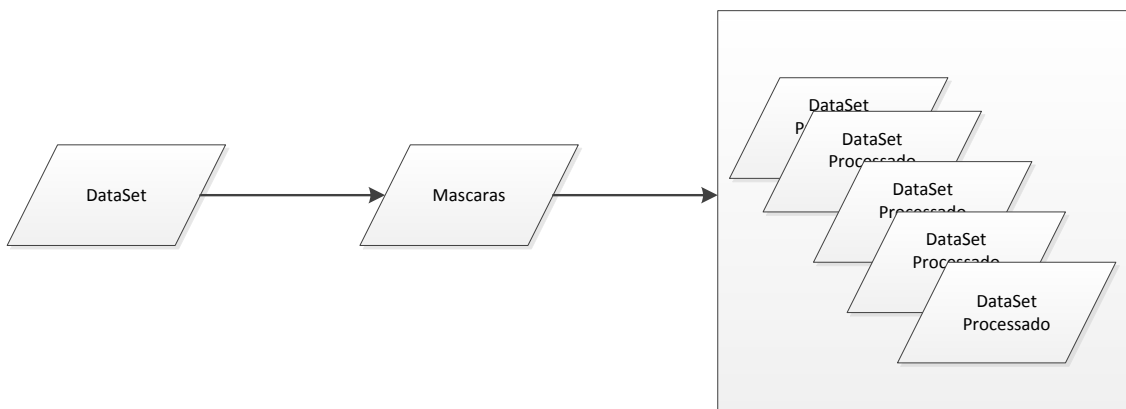


Figura 3-6: Máscaras do DataSet

3.2 Base de Dados

Para a construção da base de dados foi usada a ferramenta *MySql Workbench* que tal como o nome indica usa *MySQL*, um *database management system (DBMS)* em que se trabalha com a linguagem *SQL (Structured Query Language)*.

MySQL é um dos *DBMS* atualmente mais usados, especialmente para o ensino por ser gratuito, havendo no entanto uma versão otimizada para aplicações comerciais a qual já tem um preço associado. Existem vários *DBMS* e alguns dos mais conhecidos são *MySQL*, *Microsoft SQL Server*, *Oracle*, *MariaBD*, *PostgreSQL*, *SQLite* que funcionam com *SQL*, mas diferem na forma como é construída a *query*. A escolha do *MySql* como *database management system* a usar nesta tese, foi por já se ter usado em outros projetos, havendo mais experiência e um maior à vontade e por ser uma das *DBMS open-source* mais usadas na atualidade.

O *MySql Workbench* é uma ferramenta *open-source*, gratuita sendo usada com a *Microsoft Visual Studio* para trabalhar com *MySql*.

O uso de *MySql Workbench* tem inúmeras vantagens, tais como poder correr *queries* dentro da ferramenta para testar a sua construção e bom funcionamento, criar tabelas e os seus atributos de forma intuitiva, dispõe de opções para obtenção rápida de *queries* para tabelas específicas no formato de *Insert Statement*, *Update Statement*, *Delete Statement* e *Create Statement* ajudando na construção rápida de *queries* simples.

Para o desenvolvimento duma base de dados é recomendado que a base de dados seja normalizada para evitar repetição de informação. Normalizar a base de dados evita o desperdício da memória e facilita na atualização ou qualquer outra operação a efetuar na base de dados. Um exemplo é a Tabela 3-3, que tem dois *tuplos* com a mesma conta (A-201) que tem saldo de 900, pertencente a dois clientes diferentes. Observa-se a repetição da informação em dois atributos “*account_number*” e “*balance*”, em que no caso de se querer actualizar o saldo da conta tem de se alterar os dois *tuplos*. Caso se normalize a tabela obtêm-se duas tabelas “*conta*” (Tabela 3-4) e “*depositante*” (Tabela 3-5), neste caso se quisermos alterar o saldo é necessário actualizar só um *tuplo* e não se desperdiça memória a guardar duas vezes o mesmo saldo [17].

Outro aspeto relevante no desenvolvimento de uma base de dados é o uso de tabelas que contêm *null*, que cria alguma polémica na comunidade de profissionais que trabalham com bases de dados. A polémica existe em se poder ou não usar *null* para a construção correta de uma base de dados. Pois uma das condições da normalização de uma tabela é que esta não tenha nenhum *null* [18]. Nesta tese optou-se pelo uso de *nulls*.

Tabela 3-2: Tabela depositante

customer_id	account_number	balance
192-83-7465	A-201	900
019-28-3746	A-201	900

Tabela 3-3: Tabela conta

account_number	balance
A-201	900

Tabela 3-4: Tabela depositante normalizada

customer_id	account_number
192-83-7465	A-201
019-28-3746	A-201

O diagrama da base de dados que se desenvolveu para a plataforma *online* encontra-se na figura 3-7. Foi obtido através da ferramenta *MySQL Workbench* já referida.

O diagrama obtido é composto por várias tabelas ou entidades, distinguindo-se três tabelas pela sua importância *user*, *image*, *topic*. A tabela *image* guarda dados de três tipos diferentes de imagens *DataSet*, *DataSet Processado* e *Topic image*. No caso da tabela *user* guarda a informação relativa a um utilizador. A tabela *topic* representa um tópico de um *fórum*, ao qual estão associados comentários de utilizadores feitos nesse tópico, título do tópico, criador tópico e outros dados, mas para além disso está associado a vários conjuntos de imagens *DataSet*, *DataSet Processados* e *Máscaras*. As restantes tabelas estão associadas a estas, contendo informação adicional para o funcionamento correto da *plataforma online*.

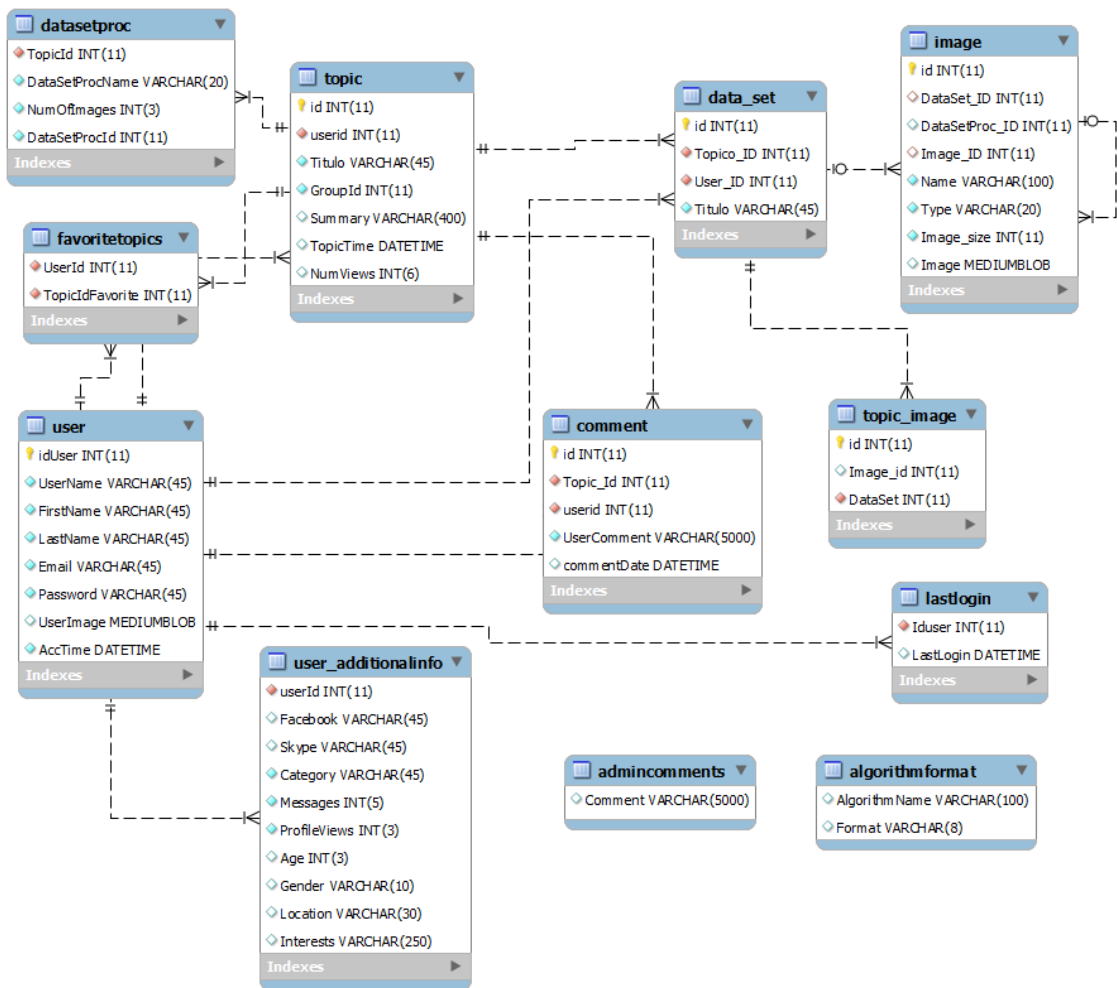


Figura 3-7: Diagrama de Entidades e Relações da Base de Dados

A tabela que se chama *image* na realidade tem vários atributos em que a maior parte das vezes se encontram *null*. Não se repartiu em várias tabelas pois esses atributos servem para identificar os vários tipos de imagens e as suas interligações, observe-se a figura 3-8.

id	DataSet_ID	DataSetProc_ID	Image_ID	Name	Type	Image_size	Image
1	1	NULL	NULL	8_icone...	jpg	5247	BLOB
2	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
3	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
4	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
5	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
6	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
7	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
8	1	NULL	NULL	Trial 00...	bmp	4195716	NULL
9	1	NULL	NULL	Trial 00...	bmp	4195716	NULL
10	1	NULL	NULL	Trial 00...	bmp	7318235	NULL
11	1	NULL	NULL	Trial 00...	bmp	4195716	NULL

Figura 3-8: Conjunto de *tuplos* de um *DataSet*

A figura referida contém um conjunto de *tuplos* que representam um *DataSet*. Todos os *tuplos* exceto o primeiro representam um *DataSet*, pois os atributos “*DataSetProc_ID*” e “*Image*” encontram-se a *null*. O primeiro *tuplo* não pertence ao *DataSet* pois contém a imagem do tópico no atributo *Image*.

Passando para um conjunto de imagens que pertencem a um *DataSet Processado* que se encontram na figura 3-9, consegue ver que o atributo *DataSetProc_ID* já se encontra preenchido com um valor, o que representa que o conjunto de *tuplos* mostrados pertence a um *DataSet Processado*.

id	DataSet_ID	DataSetProc_ID	Image_ID	Name	Type	Image_size	Image
25	1	0	NULL	Trial 00...	bmp	5490774	NULL
26	1	0	NULL	Trial 00...	bmp	5490774	NULL
27	1	0	NULL	Trial 00...	bmp	5490774	NULL
28	1	0	NULL	Trial 00...	bmp	3146806	NULL
29	1	0	NULL	Trial 00...	bmp	3146806	NULL
30	1	0	NULL	Trial 00...	bmp	3146806	NULL
31	1	0	NULL	Trial 00...	bmp	3146806	NULL
32	1	0	NULL	Trial 00...	bmp	5490774	NULL
33	1	0	NULL	Trial 00...	bmp	3146806	NULL
34	1	0	NULL	Trial 00...	bmp	3146806	NULL
35	1	0	NULL	Trial 00...	bmp	5490774	NULL

Figura 3-9: Conjunto de *tuplos* de um *DataSet Processado*

Como se pode observar só a imagem do tópico é guardada na base de dados (figura 3-8 e figura 3-9), todas as restantes imagens que pertencem aos *DataSet*, *DataSetProcessado* e *Máscaras* encontram-se armazenadas em pastas no formato *zip* numa pasta no Servidor.

A comunicação da plataforma com a base de dados é feita através de *Queries SQL*. Uma *query* é uma *string* que representa o código necessário para aceder a uma tabela para inserir, seleccionar, remover ou atualizar qualquer tipo de dados.

3.3 Processamento Digital de Imagem

A maior parte das imagens provêm de câmaras digitais ficando cada *pixel* guardado no formato *RGB*. No caso das imagens digitais é possível conter mais dados como *depth* (profundidade) que representa a distância do *pixel* da câmara virtual e o *alpha* que representa o nível de transparência.

As imagens digitais são representadas no formato de uma matriz $m * n$ encontrando-se em cada posição um *pixel*. Cada *pixel* é definido por um nível de cinzento ou noutros casos um *array* de três posições (*RGB format*) em que da média $\left(\frac{R+G+B}{3}\right)$ ou da mistura das três cores obtém-se o nível de cinzento (*grayscale*), o qual também é denominado intensidade. O formato convencional é o *RGB*, cada *pixel* é definido por um inteiro que vai desde 0 a 255 que é a intensidade da luz. O conjunto de todos os valores de vermelhos correspondentes a todos os píxeis de uma imagem chama-se canal vermelho, cada valor é um inteiro de 8 bits é o formato mais comum de se usar chamando-se canal vermelho de 8 bits.

As imagens são guardadas em vários formatos de armazenamento. A principal diferença entre os diferentes formatos é se existe compressão ou não, e se houver se a compressão é com perdas ou sem perdas. Entende-se por compressão sem perdas aquela em que são removidos bits através da identificação e remoção de redundância estática. No caso com perdas é possível nalguns casos parecer sem perdas, ao não ser distinguida pelo olho humano a imagem comprimida da não comprimida, havendo na realidade perdas visíveis pelo computador mas não pelo sistema visual de um ser humano. Para isso aproveitam-se as limitações que o sistema visual do ser humano tem, nomeadamente ser menos sensível na deteção de mudanças suaves entre cores.

3.3.1 Formatos de Imagens

O *JPEG* (*Joint Photographic Experts Group*) é um formato muito popular na maior parte das câmaras fotográficas por produzirem as imagens nesse formato, sendo um *standard* em imagens naturais por esse motivo. É um formato com perdas, pois a sua compressão é feita através da divisão de uma imagem em vários blocos pequenos e comprimindo a informação guardada em cada bloco. Fazendo *zoom* numa imagem em formato *JPEG*, podem ser visíveis esses blocos e é visível a má qualidade com que a imagem fica, dando a impressão de ver a imagem *pixelizada*. Um exemplo desta situação é apresentado na Figura 3-10 em que a primeira imagem *JPEG* é sem *zoom*. Na segunda imagem que se obteve fazendo *zoom* da primeira, já se nota uma grande perda de qualidade da imagem, sendo visíveis os blocos que aparecem quase com a forma de um quadrado *pixel*.



Figura 3-10: Imagens JPEG

Por isso é difícil o seu uso em comparação de imagens, pois torna-se impossível verificar se as duas imagens são realmente iguais ou se as diferenças provêm simplesmente do algoritmo de *JPEG* ter escolhido diferentes valores para cada imagem criando a ideia de serem imagens diferentes.

GIF (*Graphics Interchange Format*) é um formato de imagem comum quando o armazenamento de imagens era crítico e os *scanners* e as câmaras digitais eram raros. A razão para esse fato é que contém uma tabela de cores limitada a 256 cores sendo uma imagem *GIF* limitada às 256 cores definidas nessa tabela. *GIF* é o formato ideal para imagens que tenham poucas cores, como nas imagens de *logótipos* e *ícones* simples. Para imagens naturais que têm uma gama grande de cores, perdem quase toda a qualidade da imagem em formato *GIF*, sendo por isso um formato geralmente com perdas nesta situação.

BMP (Bitmap Image File) é um formato capaz de armazenar imagens. Em que os *píxeis* estão definidos por uma profundidade de cor, isto quer dizer que cada *píxel* é representado por 1, 2, 4, 8 ou 24 *bits*. Sendo as imagens de 24 *bit* definidas pelo modelo de cor *RGB* e as restantes por mapas de cores. A compressão é feita pela capacidade de armazenamento de *bits* por *píxel*, sendo o 24 *bits* a melhor representação de uma imagem real, pois é a que contém a maior gama de cores. Quanto menor for o número de *bits* por *píxel* maior será a sua compressão e pior será a sua qualidade (Figura 3-11).



Figura 3-11: da esquerda para a direita 1bit/pixel, 2bit/pixel, 24 bit/pixel.

O *TIFF (Tagged Image File Format)* é um formato em que a descrição de cada *píxel* é composto por cinco valores. Três são valores inteiros que vão de 0 a 255 e que são as intensidades das cores vermelha, verde e azul. Um é o valor z que representa a profundidade e por último o valor inteiro de 16 *bits*, que representa a identificação de um objeto. O parâmetro z que representa o conceito de profundidade ou camadas. As imagens no formato *TIFF* têm a representação mais fiável do mundo real, sendo por isso o ideal para qualquer processamento digital. É adequado para a comparação de duas imagens, porque para além da comparação da intensidade de cada pixel, poder-se-á também comparar os valores de profundidade [19].

3.3.2 Modelos de Cor

Modelo de cor também referido como espaço de cor ou sistema de cor, serve para especificar em que formato estão definidas as cores de uma imagem. Serve para saber como processar cada imagem ou simplesmente mostrar a imagem num visor. Como foi referido o modelo de cor mais usado é o *RGB 24 bits* que é o usado nesta tese. Deve-se à sua boa capacidade de representar uma imagem real, pois este modelo representa 8 *bits* por canal ou seja 24 *bits* pelos três canais, obtendo-se um total de $(2^8)^3=2^{24}=16.777.216$ cores, que se encontram representadas no cubo com os valores normalizados [0,1] (Figura 3-12).

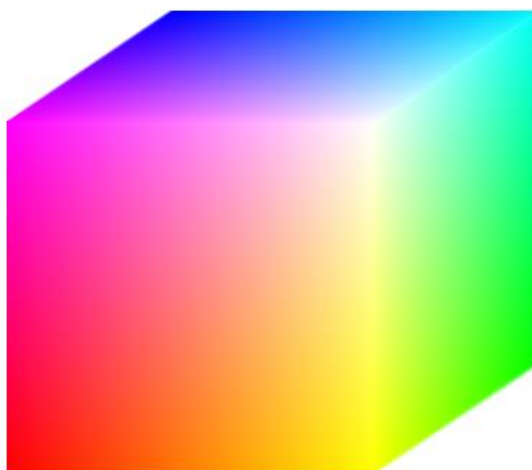


Figura 3-12: Cubo do modelo de cor RGB

Sendo a cor preta definida pelas coordenadas $(0,0,0)$ representadas no cubo e que é a origem que não se encontra visível na Figura 3-7. A cor branca encontra-se nas coordenadas $(1,1,1)$ no cubo, correspondendo às coordenadas $(255,255,255)$ não normalizados e que já é visível no cubo, é a ponta mais próxima sendo esbranquiçadas as cores ao seu redor.

Os outros dois modelos de cor mais usados são CMY ou CMYB e HSI.

Em que *CMY* representa *Cyan*, *Magenta* e *Yellow*. É o formato usado pelas impressoras que representa as propriedades dos pigmentos que são depositados no papel durante a sua impressão. Por exemplo no caso do pigmento ciano que quando é iluminado por uma luz branca não reflete nenhuma luz vermelha de volta.

HSI é um modelo de cor representado por *Hue*, *Saturation* e *Intensity* em que é separado o tom da cor e a saturação, da intensidade [1].

3.3.3 Máscaras

As *máscaras* usadas na tese servem para indicar os *píxeis* de interesse, ignorando os *píxeis* fora da *máscara*. Também entram nos cálculos quando queremos obter “*Total-dePíxeis*” que representa o total de *píxeis* de interesse (Figura 3-13).

3.3.4 Imagens em Silverlight

O processamento digital na *Plataforma online*, será efetuado na visualização de um tópico e na escolha de vários *DataSet Processados*, em que se irão comparar duas imagens e retirar os valores da sua comparação. Para esse processamento ir-se-á usar o modelo de cores no formato *pARGB* (*premultiplied Alpha, Red, Green e Blue*) pois é o formato definido pela classe *WriteableBitmap* que se encontra na tecnologia em que se trabalhou (*Silverlight*). A classe referida é usada para imagens a que se queira aceder diretamente aos *píxeis*, através da propriedade *Pixels* que é um array de dimensão equivalente ao número de *píxeis* da imagem. Cada *pixel* está no formato referido *pARGB*, que é um inteiro de 32 *bits*, sendo a ordem dos *bits* a começar dos mais significativos para os menos significativos. Os primeiros 8 *bits* correspondem à transparência do *pixel* (mais conhecido por *alpha*), seguido por 8 *bits* que correspondem à intensidade de vermelho, seguido pelos 8 *bits* que correspondem à intensidade do verde. Finalizando os últimos 8 *bits* que representam a intensidade do azul. Uma amostra de código em *C#* de como se obter a cor de um *pixel* de uma imagem.

```
WriteableBitmap wbmp = new WriteableBitmap(Source);
int Pixel = wbmp.Pixels[i];

int B = (int)(Pixel & 0xFF); Pixel >>= 8;
int G = (int)(Pixel & 0xFF); Pixel >>= 8;
int R = (int)(Pixel & 0xFF); Pixel >>= 8;
int A = (int)(Pixel);

Color color = Color.FromArgb((byte)A, (byte)R, (byte)G, (byte)B);
```

3.3.5 Tópicos

Na plataforma desenvolvida, o processamento de imagem ocorre na visualização de um tópico, onde é feita a comparação entre imagens. Na comparação são obtidos alguns valores nomeadamente: *Verdadeiro Positivo* (VP); *Verdadeiro Negativo* (VN); *Falso Positivo* (FP); *Falso Negativo* (FN). Destes valores calcula-se a *Sensibilidade* (SS), a *Especificidade* (SE) e o *Kappa* descritos na tabela 3-6 [5].

Tabela 3-5: Variáveis e Fórmulas na comparação entre duas imagens

Verdadeiro Positivo (VP)	É o número de pixéis marcados em ambas as imagens.
Verdadeiro Negativo (VN)	$VN = TotaldePixeis - FP - FN - VP$
Falso Positivo (FP)	É o número de pixéis que se detetou na imagem e que não se devia ter detetado
Falso Negativo (FN)	$FN = total - VP$
Sensibilidade (SS)	$\frac{VP}{VP + FN}$
Especificidade (SP)	$\frac{VN}{VN + FP}$
Pr(a)	$\frac{VP + VN}{Total_Pixeis}$
Pr(e)	$\frac{(VP + FP) * (VP + FN)}{Total_Pixeis^2} + \frac{(VN + FN) * (VN + FP)}{Total_Pixeis^2}$
Kappa	$\frac{Pr(a) - Pr(e)}{1 - Pr(e)}$

Para o cálculo de VN, FN, Pr(a) e Pr(e) são necessárias as variáveis “*TotaldePixéis*”, “*Total*” e “*Total_Pixéis*”, sendo “*TotaldePixéis*” a variável que representa o número total de *pixéis* de interesse (a preto) definidos pela *máscara* Figura 3-13.

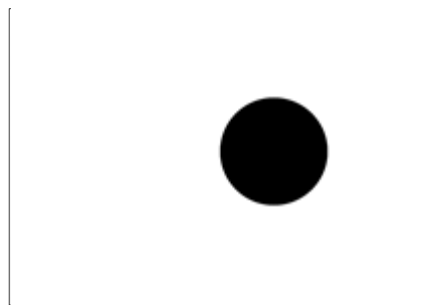


Figura 3-13: A preto são os *pixéis* da *máscara* a indicar a área de interesse

“*Total*” é a variável que representa o número total de *pixéis* detetados na imagem ideal (a imagem com que as restantes irão ser comparadas), em que esses são os que vão ser comparados com os *pixéis* das restantes imagens Figura 3-14.

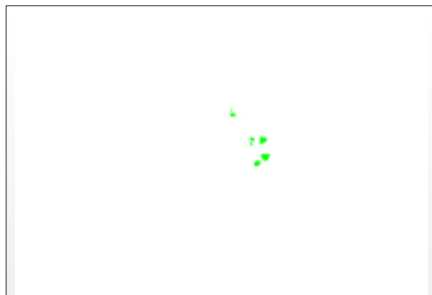


Figura 3-14: Os *pixéis a verde* são os *pixéis* detetados numa imagem ideal

“*Total_Pixéis*” é a variável que representa o número total de *pixéis* de uma imagem de um tópico (num tópico todas as imagens do *DataSet*, *DataSet Processado* e *Máscaras* têm o mesmo tamanho).

De referir que de todos os valores obtidos na comparação, são só mostrados a *Sensibilidade*, a *Especificidade* e o *Kappa*, por serem os valores de maior interesse estatístico. A partir desses valores será possível avaliar o grau de eficácia dos algoritmos.

3.4 Instalação (Deploy)

Durante a criação da plataforma, esta foi sempre testada localmente. Os erros detetados foram sendo corrigidos ao longo do seu desenvolvimento. Posteriormente a plataforma foi colocada *online* de modo a ser acessível a qualquer utilizador. Para esse efeito o projecto foi publicado (*publish*), sendo transferidos para o Servidor definido todos os ficheiros essenciais. Estes últimos são: uma pasta *bin* contendo todas as *dll's* necessárias; uma pasta *ClientBin* que contém o ficheiro no formato *xap*; a pasta *ServerImages* que conterà todas as imagens; uma pasta *Service* que tem o ficheiro *WCF Web Service*; e na pasta principal ficam os restantes ficheiros *Generic Handlers*, *Web.config* entre outros.

Ao aceder à plataforma online detetou-se que ocorria uma falha, mas não estava a retornar nenhum erro, apenas era retornada uma mensagem genérica “*The remote Server returned an error: Not Found*” impossibilitando a correção do erro. Pesquisando através do Google, encontrou-se uma ferramenta chamada *fiddler* (que é um *proxy* gratuito para fazer *debug* na *web* ao receber todas as mensagens de *HTTP* de tráfico entre o computador e a *internet*) e com o seu uso conseguiu-se identificar o erro que estava a ocorrer. Para visualizar a mensagem de erro escolheu-se a opção *WebView* ou *Raw*. A mensagem de erro que o servidor estava a retornar provinha de uma falha de configura-

ção do próprio servidor *IIS 7*. Para a correção dessa falha teve que se passar a pasta principal *SIDAC* de uma diretoria virtual para uma aplicação.

Com este erro corrigido ainda se detetaram mais três erros através do *fiddler* que indicavam a falta de três dll's "*System.ServiceModel.DomainServices.Hosting*", "*MySql.Data*" e "*System.ServiceModel.DomainServices.Server*" que não se encontravam no servidor. A solução foi copiar os três dll's em falta para a pasta bin no servidor.

4 Validação

Irá ser feita uma descrição da *plataforma online*, tal como o seu correto uso na criação de um novo caso de estudo. É possível aceder em qualquer *browser* (*Internet Explorer, Firefox, Safari, Chrome*) e de qualquer sistema operativo (*Mac OS, Windows, Linux*).

A plataforma *SIDAC* divide-se em duas componentes principais. A primeira componente será responsável pelo aspeto da plataforma e pelo processamento de dados, que se encontra no lado do cliente. A segunda componente é responsável pelo armazenamento de dados e pelo armazenamento de imagens, que se encontra no lado do servidor. As duas componentes comunicam-se através da *web service*.

A *plataforma online* desenvolvida terá uma Página Principal referida como “*Main Page*” e um menu principal (Barra de Navegação) que será essencialmente composto por quatro opções e dois botões. As opções são “*Main Page*”, “*About us*”, “*Create New Topic*” e “*View Topics*” e os botões são “*Sign In*” e “*Create Account*”. Na figura 4-1 é apresentada a Barra de Navegação.



Figura 4-1: Barra de Navegação

Qualquer escolha que venha a ser efetuada por um utilizador ir-se-á refletir na *frame* que se encontra na figura 4-2.

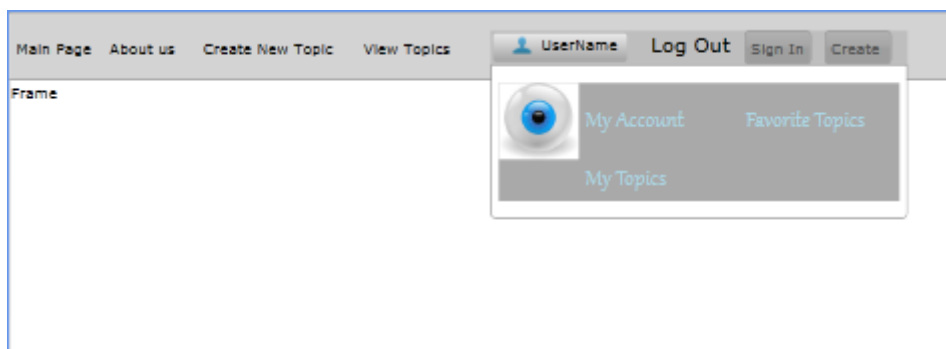


Figura 4-2: Frame

4.1 Home Page

A página principal que é visível na *frame* (figura 4.3) será composta por três secções; a primeira secção contém um conjunto de míni filmes a explicar o uso correto da *plataforma online*. A segunda secção é um conjunto de avisos provenientes do *Administrador* referentes à plataforma. A terceira secção é o conjunto de cinco tópicos com maior número de visualizações.

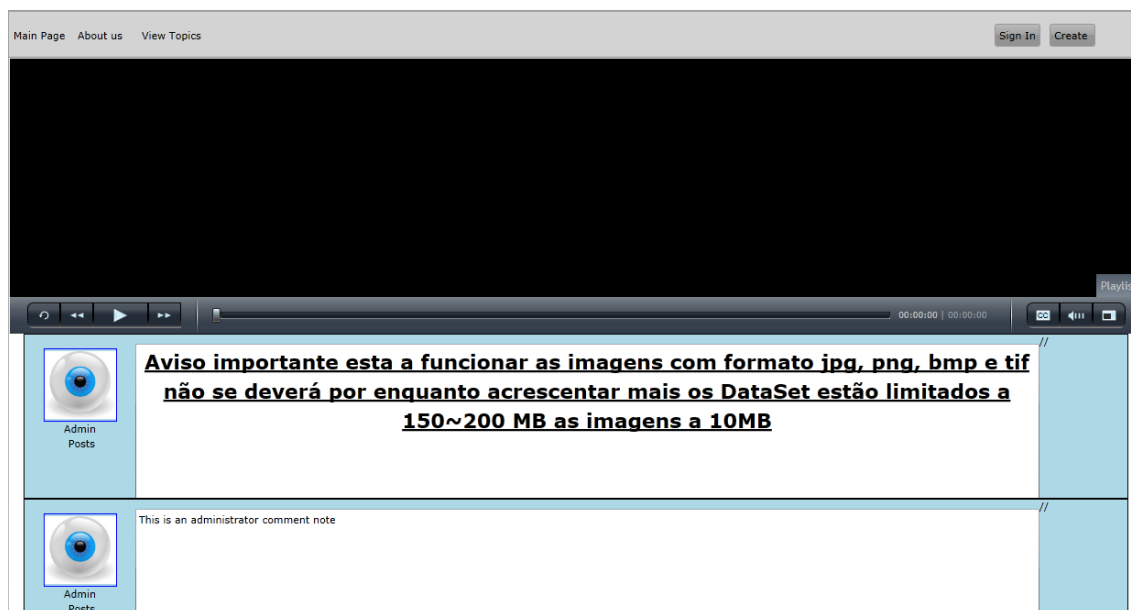


Figura 4-3: Home Page

Sempre que o *Administrador* faz *login* é disponibilizada uma ferramenta para inserir comentários. Ao inserir um novo comentário vai-se criando uma lista de avisos do *Administrador* para todos os utilizadores da plataforma, visível na Figura 4-4.

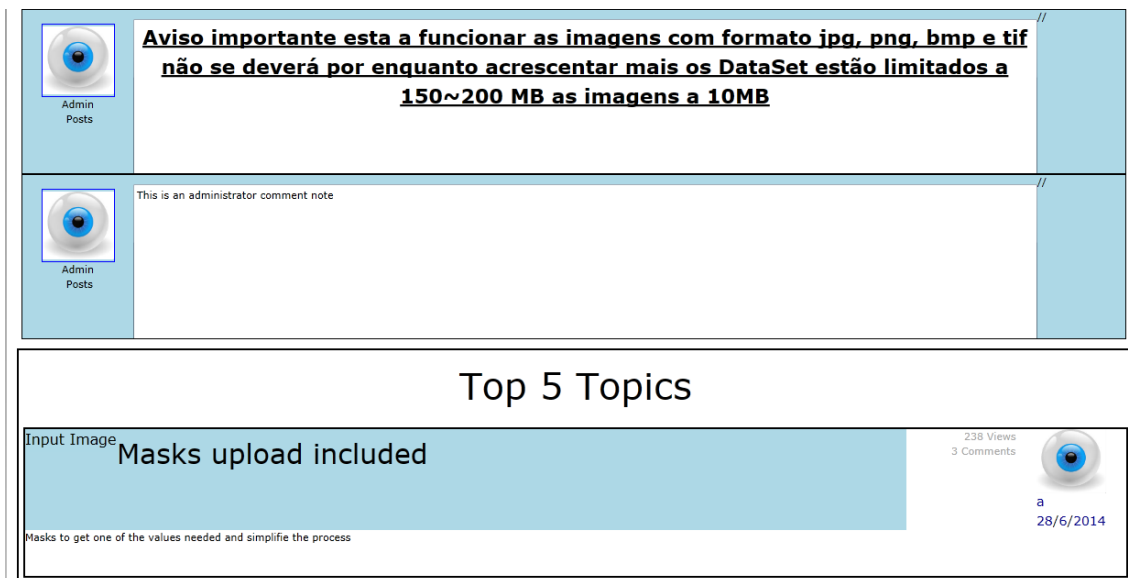


Figura 4-4: Avisos do Administrador

4.2 About Us

É a página onde é disponibilizado aos utilizadores informação sobre quem desenvolveu a *Plataforma online* e em que âmbito foi desenvolvida, que se encontra na figura 4-5.

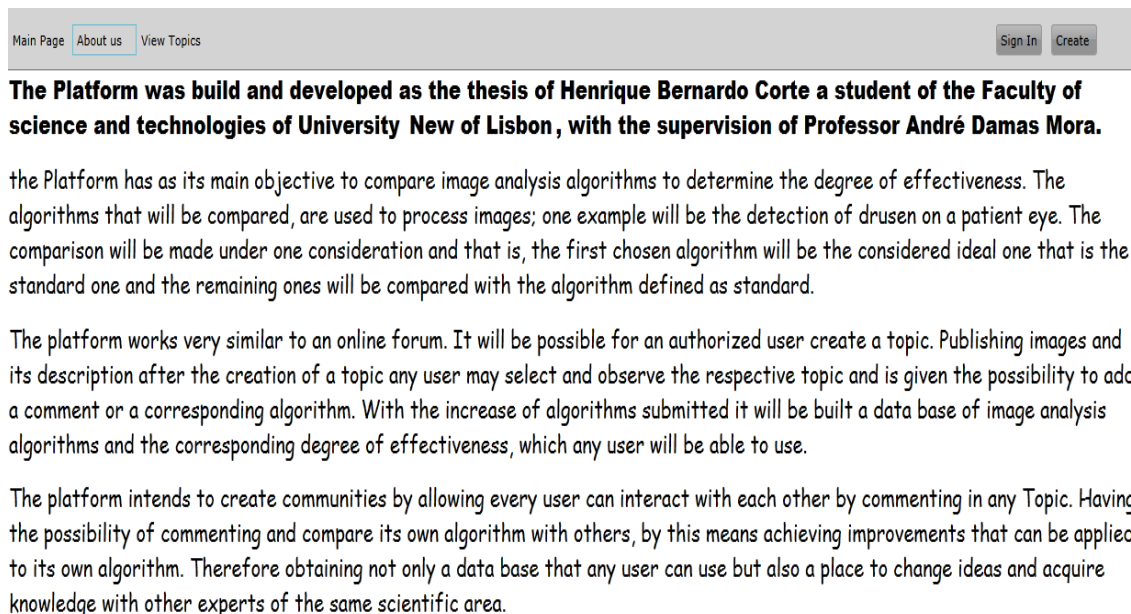


Figura 4-5: About Us

4.3 Create Account

Para um utilizador ter acesso a todas as funcionalidades na plataforma, este terá que ter uma conta e efetuar o respetivo *login*. Embora, não tendo conta ou sem ter feito o login seja possível visualizar alguns elementos da plataforma, as restantes funcionalidades são restritas e por isso encontram-se bloqueadas. Para a criação de uma nova conta basta selecionar o botão *Create*, aparecendo a página que se vê na Figura 4-6.



The screenshot shows a web interface for creating an account. At the top, there is a navigation menu with links for 'Main Page', 'About us', and 'View Topics'. To the right of the navigation menu are two buttons: 'Sign In' and 'Create'. Below the navigation menu is the 'Create Account' form. The form contains six input fields, each with a label to its left: 'User Name', 'First Name', 'Last Name', 'Email Address', 'Password', and 'Confirm Password'. At the bottom left of the form area, there is a 'Create Account' button.

Figura 4-6: *Create Account*

Como é possível visualizar, para a criação da conta, é necessário o preenchimento de alguns dados. O preenchimento de todos os campos é obrigatório e em caso da confirmação da *password*, se estiver errada será visível uma mensagem de erro.

Durante este processo e por motivo de segurança a *password* inserida pelo utilizador é codificada antes da criação da conta do utilizador na base de dados.

4.4 Login

Para efetuar o *login* basta pressionar o botão *login* que se encontra ao lado do *Create*, aparecendo a janela na Figura 4-7.

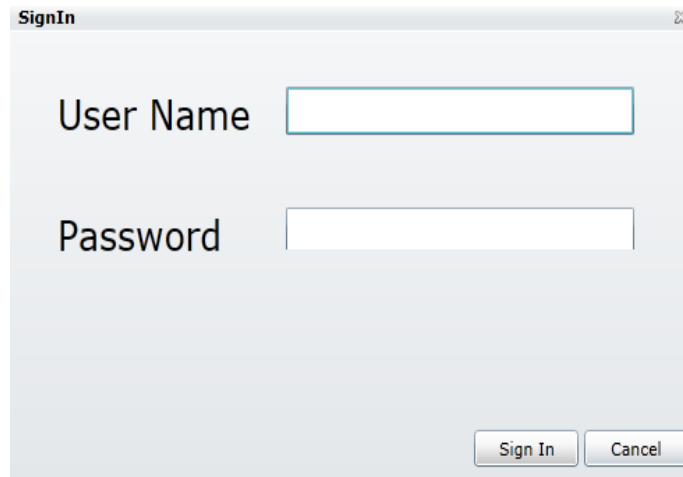
The image shows a standard Windows-style dialog box titled "SignIn". It has a light gray background and a title bar with the text "SignIn" and a close button icon. Inside the dialog, there are two labels: "User Name" and "Password", each followed by a rectangular text input field. At the bottom right of the dialog, there are two buttons: "Sign In" and "Cancel".

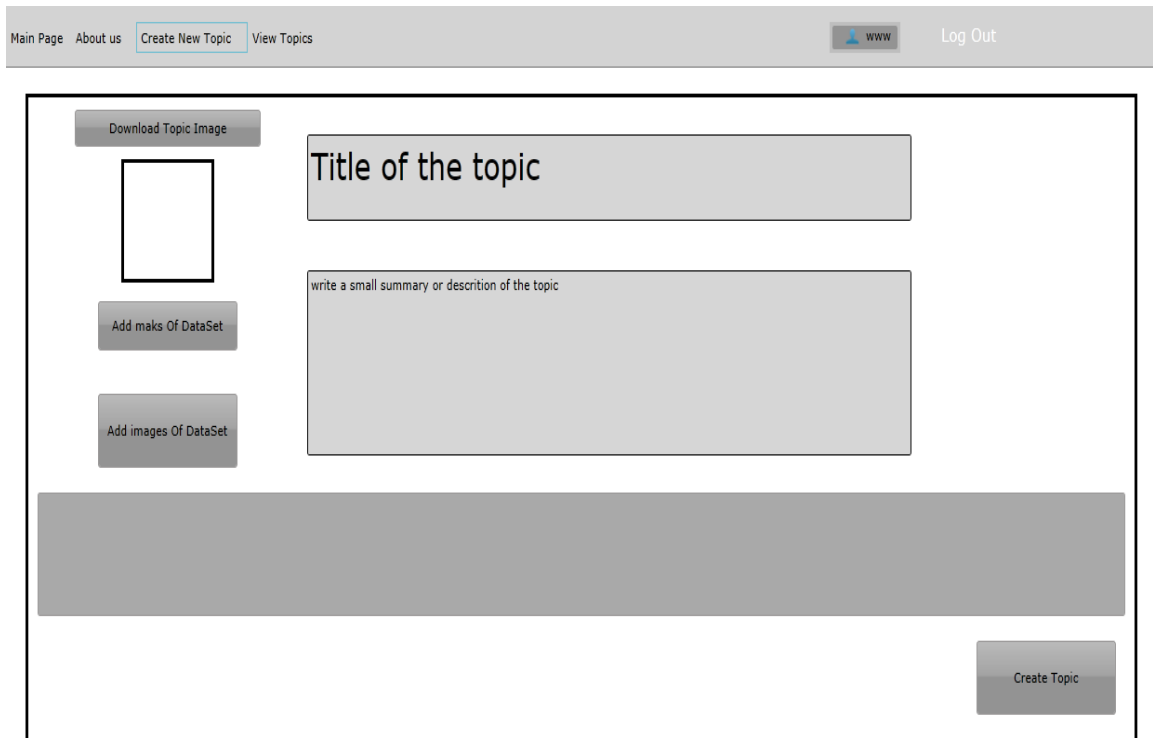
Figura 4-7: *User Login*

Preenchendo corretamente os campos completar-se-á o *login* aparecendo o *user-name* respectivo. Caso um dos dados tenha sido incorretamente inserido será mostrada uma mensagem indicando que um dos campos foi mal preenchido, mantendo a janela de *signIn* visível.

A *password* inserida é codificada de modo a comparar com a *password* vinda da base de dados que já se encontra codificada. Deste modo o *Administrador* não tem acesso a nenhuma *password* original e mesmo que esta seja interceptada na comunicação não será identificada por se encontrar codificada.

4.5 Create New Topic

Para criar um novo tópico, o utilizador terá que previamente ter criado uma conta e efetuado o *login*, caso contrário não será visível a opção de criação de um novo tópico. Após a escolha da criação de um novo tópico será visível a página na figura 4.8.



The screenshot shows a web interface for creating a new topic. At the top, there is a navigation bar with links for 'Main Page', 'About us', 'Create New Topic' (highlighted), and 'View Topics'. On the right side of the navigation bar, there is a user profile icon labeled 'www' and a 'Log Out' button. The main content area is enclosed in a black border and contains the following elements:

- A 'Download Topic Image' button at the top left.
- A rectangular placeholder for a topic image.
- An 'Add masks Of DataSet' button below the image placeholder.
- An 'Add images Of DataSet' button below the mask button.
- A text input field labeled 'Title of the topic'.
- A larger text area with the placeholder text 'write a small summary or description of the topic'.
- A large grey rectangular area at the bottom of the form, likely a placeholder for a preview or additional content.
- A 'Create Topic' button at the bottom right.

Figura 4-8: Create New Topic

Para a criação de um novo tópico será pedido ao utilizador a escolha de uma imagem que identifique o tópico, através do botão “*Download Topic Image*”. A escrita de um título e uma descrição curta sobre o tópico não serão obrigatórios mas recomendados. O *upload* do conjunto de imagens originais (*DataSet*) é feito através do botão “*Add images of DataSet*” e o *upload* do respetivo conjunto de *máscaras* no botão “*Add mask of DataSet*” como é visível na figura 4.8.

O formato disponível para o *upload* de imagens e de *máscaras* de um *DataSet* está definido na Tabela 3-2. O limite por imagem é de 10 MB e o número máximo de imagens para um *DataSet* é de 100 com um limite máximo de 250 MB.

4.6 View Topics

Para a visualização dos tópicos existentes basta carregar na opção “*View Topics*” do menu principal, sendo visível na Figura 4-9 uma parte parcial da Página que contém uma lista de tópicos, com apenas 2 tópicos.

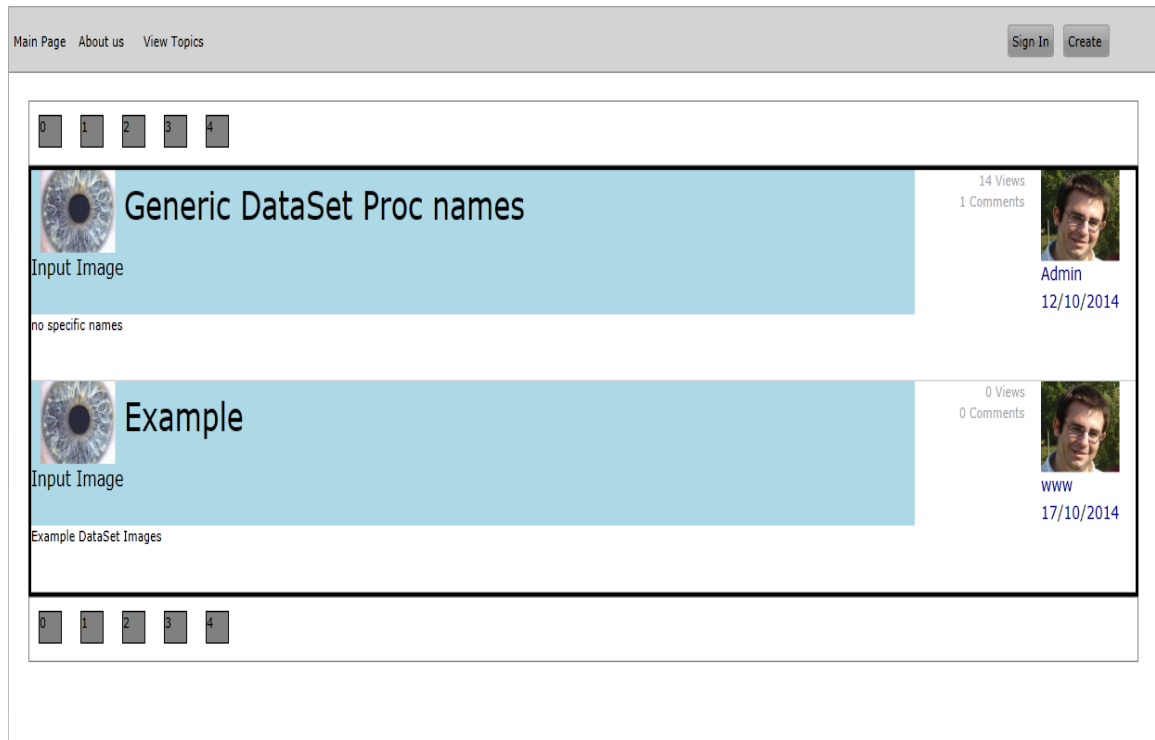


Figura 4-9: View Topics

Os números apresentados no canto superior esquerdo são referentes à paginação dos resultados, sendo apresentados no máximo dez tópicos por cada página. Para ajudar na procura de um tópico que se deseja visualizar é mostrado o Título do tópico, uma pequena descrição do tópico, a imagem do tópico, o *username* do utilizador que criou o tópico, a data da criação do tópico, o número de comentários que se encontram no tópico e é também disponibilizado o número de vezes que foi visualizado.

O número de visualizações, corresponde ao número de vezes que o respetivo tópico foi selecionado, não tendo em conta se foi um utilizador com *login* efetuado ou por um utilizador não identificado, nem se foi feito pelo mesmo *PC* o acesso de repetidas visualizações. O *username* do utilizador que criou o tópico é um *hyperlink*, que pressionado irá mostrar dados adicionais do criador do tópico caso este os tenha disponibilizado (*email*, *facebook* entre outros).

4.7 Seleção de um tópico

Ao selecionar um tópico será mostrada a página da Figura 4-10, sendo apresentadas as imagens do *DataSet* numa caixa que contém uma barra lateral para auxiliar na visualização de todas as imagens.

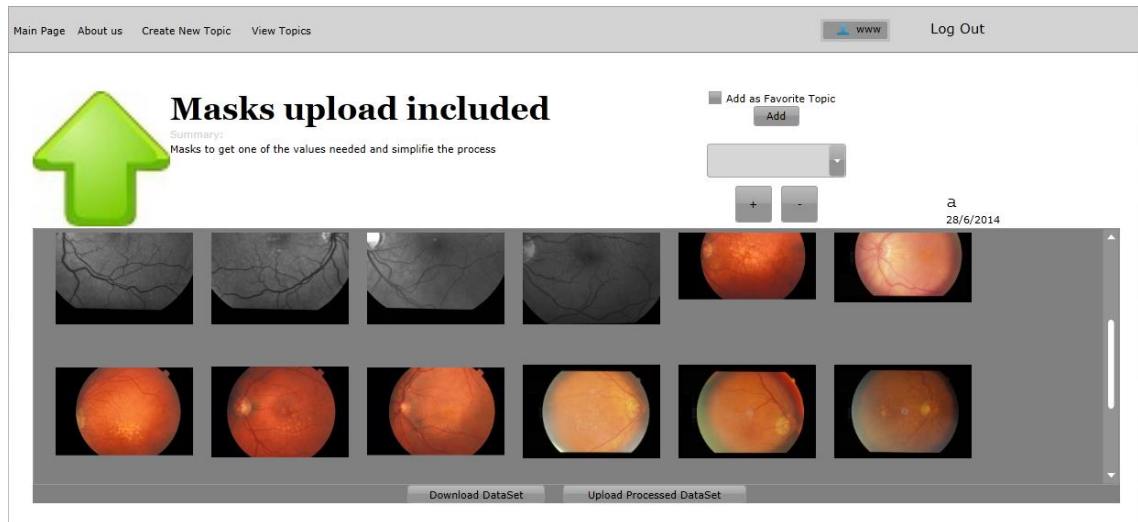


Figura 4-10: Topic selected

Conterá alguns dados informativos que já se referiram no “*View Topics*”, sendo possível também adicionar ou remover o tópico de uma lista de tópicos favoritos do utilizador. Para um utilizador que queira testar o seu algoritmo neste conjunto de imagens, ele tem a hipótese de fazer o *download* do *DataSet* para o seu computador, processar as imagens e em seguida fazer o *upload* do seu *DataSet Processado* para ser incluído neste tópico. Estas opções encontram-se bloqueadas, exceto para utilizadores que tenham efetuado um prévio *login*. Os *DataSet Processados* que tenham sido adicionados ao tópico aparecem numa *ComboBox*. Podem ser adicionados para visualização e comparação através da sua seleção e pressionando o botão *add*. Na figura 4-11 encontra-se uma imagem do *DataSet*, a respetiva *máscara* e as correspondentes imagens processadas.

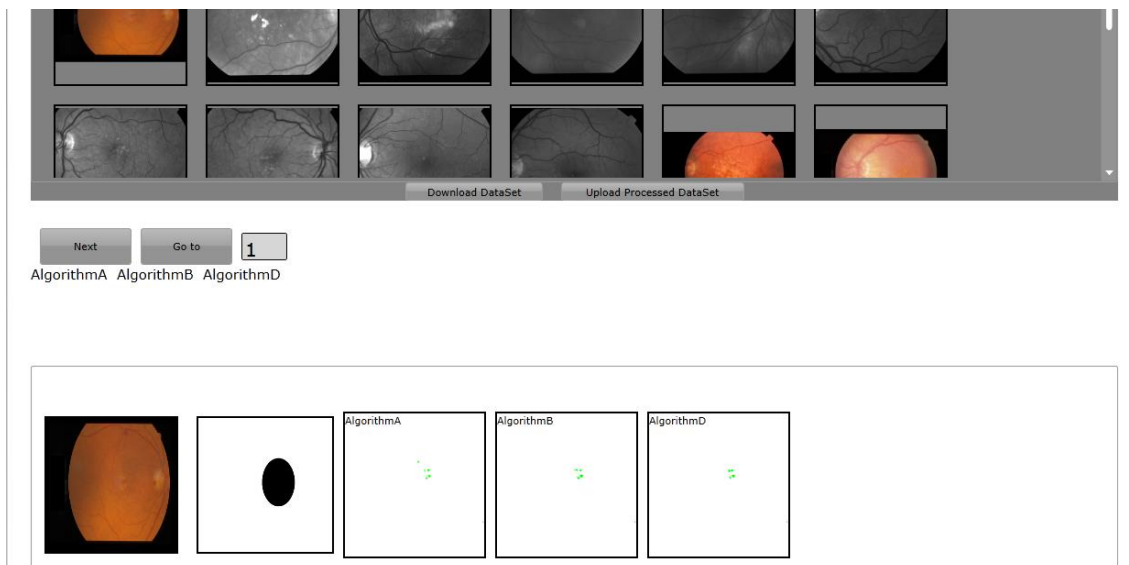


Figura 4-11: DataSet, Máscara e Imagens processadas

Sendo visíveis mais abaixo, os valores retirados das comparações das imagens processadas na figura 4-12.

	DataSetProc Name	sensitivity	specificity	Kappa Value
▶	AlgorithmA	Ideal Case	Ideal Case	Ideal Case
	AlgorithmB	0.791	0.992	0.064
	AlgorithmD	0.704	0.992	0.064

Figura 4-12: Resultados da comparação

Por fim estão os comentários que se encontram no fim da página como é visível na Figura 4-13. Estes comentários servem como meio de comunicação entre os vários utilizadores que tenham selecionado um tópico, sendo o principal objetivo escrever-se sobre os métodos de processamento que foram usados nos *DataSet Processados*, visando criar uma comunidade de investigadores que trabalhe sobre um tópico específico.

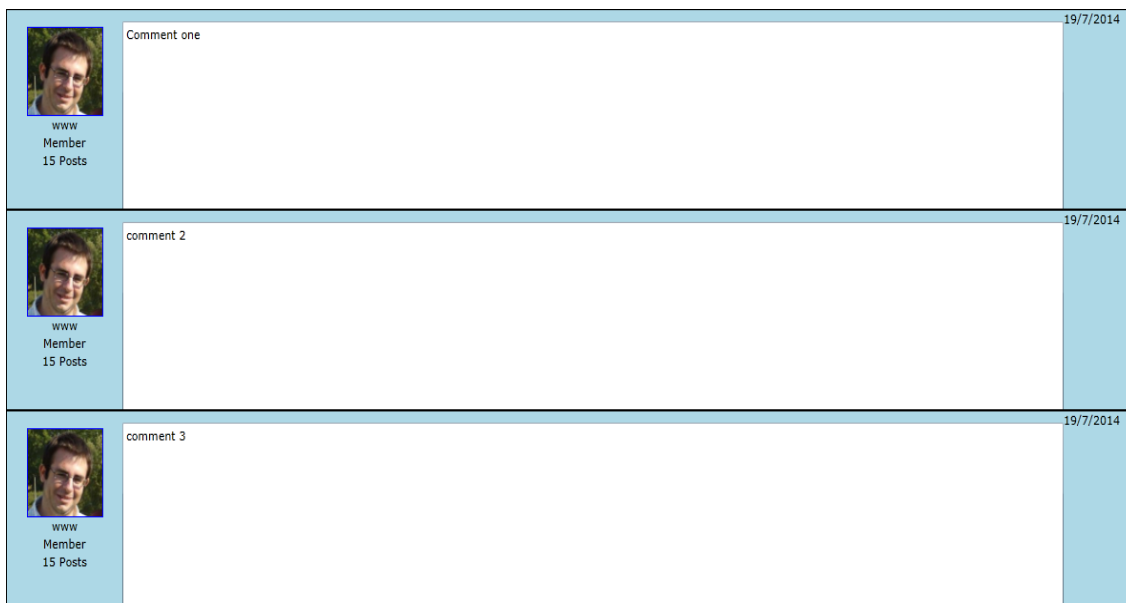


Figura 4-13: Comentários de utilizadores

4.8 Account

Ao fazer-se *login* é possível entrar na própria conta, visualizar os tópicos favoritos e os tópicos criados (Figura 4-14).

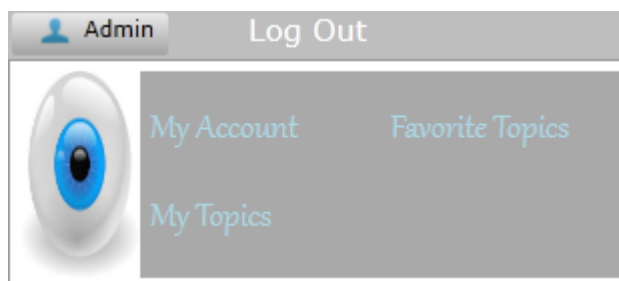


Figura 4-14: Menu de Utilizador

Ao entrar na conta Figura 4-15, são disponibilizados os dados do utilizador que foram introduzidos na criação da conta, assim como os dados extra que podem ser preenchidos pelo utilizador caso este deseje disponibilizar mais dados. Pode associar-se uma imagem à conta (de 300 kb no máximo) carregando sobre a imagem atual da conta. Para adicionar mais dados basta selecionar o “*Edit Account*”. Os dados adicionais que se poderão fornecer são *facebook*, *skype*, idade, localidade, sexo e interesses. Na conta po-

derá ver-se o último login feito que será útil para outros utilizadores verem se a conta está ativa ou não.

The screenshot shows a user profile page for a user named 'Admin'. The page has a navigation bar at the top with links for 'Main Page', 'About us', 'Create New Topic', and 'View Topics'. The user's name 'Admin' is displayed in the top right, along with a 'Log Out' button. The profile information includes a member since date of 30 August 2014 and a last login of 1 of September at 16:51. The 'User Information' section lists details such as User Name (Admin), First Name (H), Last Name (C), Email (C@), Facebook (Unknown), and Skype (Unknown). The 'General Information' section shows User Category (Member), Messages (0), Profile Views (0), Age (Unknown), Gender (Unknown), Email (C@), Location (Unknown), and Interests (Unknown). On the right side, there are buttons for 'Friends' (showing 0) and 'Latest Visitors'.

User Information	
User Name	Admin
First Name	H
Last Name	C
Email	C@
Facebook	Unknown
Skype	Unknown

General Information	
User Category	Member
Messages	0
Profile Views	0
Age	Unknown
Gender	Unknown
Email	C@
Location	Unknown
Interests	Unknown

Figura 4-15: Dados da Conta de um utilizador

Ao seleccionar-se “*My Topics*” irá aparecer a *página* da Figura 4-16 ou “*Favorite Topics*” ver-se-á uma lista semelhante à “*View Topics*” a única diferença é que no “*My Topics*” são todos os tópicos criados pelo utilizador da conta e “*Favorite Topics*” são todos os tópicos que foram adicionados pelo utilizador como tópicos de interesse, enquanto “*View Topics*” são todos os tópicos da plataforma como já foi referido.

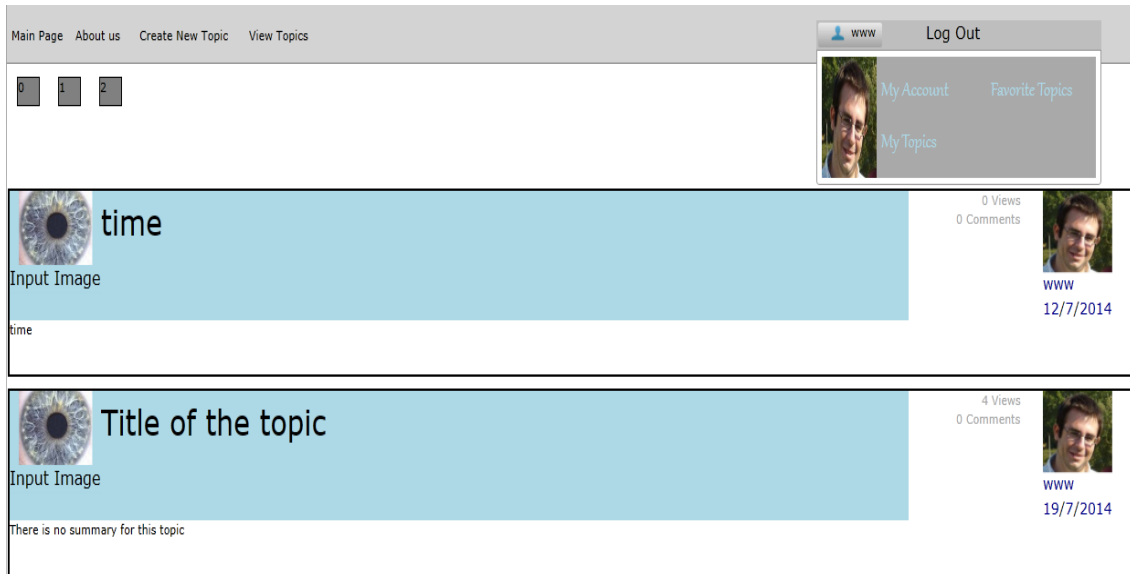


Figura 4-16: My Topics (tópicos criado pelo utilizador que fez login)

4.9 Análise de Resultados

Criou-se um tópico com uma amostra de vinte e duas imagens de retina no formato *BMP* que encontram-se na Figura 4-17.

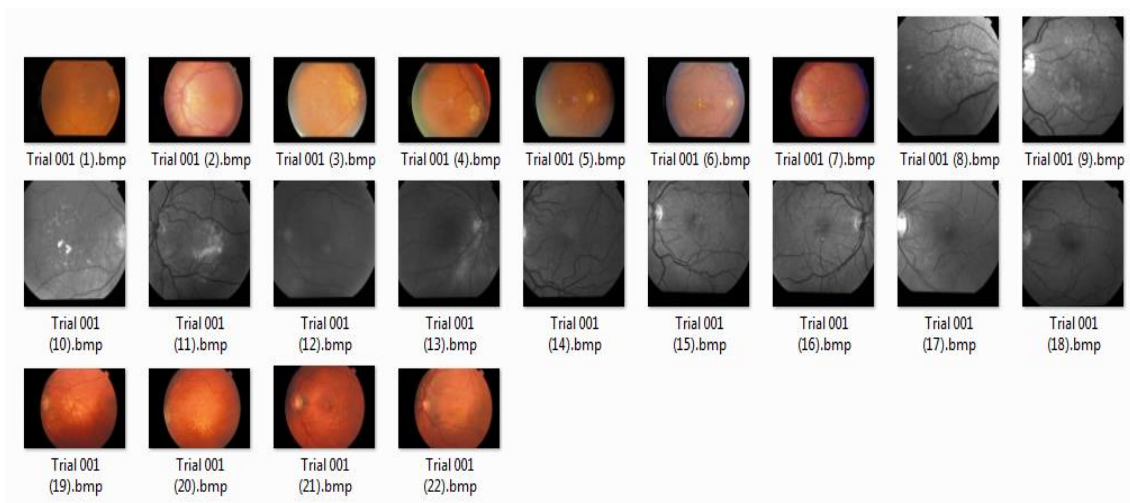


Figura 4-17: Todas imagens que pertecem ao *DataSet* criado.

A esse *DataSet* associou-se posteriormente um conjunto de oito *DataSet Processados*. Estes *DataSets Processados* estão identificados pelos nomes dos respectivos autores. Escolheram-se duas imagens de retina para fazer a análise de resultados. Escolheu-se a primeira e a décima primeira imagem, cujos nomes são: “*Trial 001(1)*” e “*Trial 001(10)*”. Está disponibilizada a lista dos nomes das imagens na Figura 4-18.

Name	
▶ Trial 001 (1)	
Trial 001 (10)	
Trial 001 (11)	
Trial 001 (12)	
Trial 001 (13)	
Trial 001 (14)	
Trial 001 (15)	
Trial 001 (16)	
Trial 001 (17)	Trial 001 (3)
Trial 001 (18)	Trial 001 (4)
Trial 001 (19)	Trial 001 (5)
Trial 001 (2)	Trial 001 (6)
Trial 001 (20)	Trial 001 (7)
Trial 001 (21)	Trial 001 (8)
Trial 001 (22)	Trial 001 (9)

Figura 4-18: Lista dos nomes das 22 imagens do *DataSet*, *DataSet Processado* e *máscaras*.

A Primeira imagem de retina escolhida para testar e obter resultados, é a primeira imagem da lista na Figura 4-18 que tem o nome “*Trial 001 (1)*”, sendo visível na Figura 4-19 com a respectiva *máscara* e as respectivas imagens dos *DataSet Processados* adicionados para comparar. Foi escolhido o *DataSet Processado* “*AlgorithmA*” como o *DataSet Processado* ideal.



Figura 4-19: Primeira imagem do *DataSet*, a *máscara* e imagens dos *DataSet Processados*.

Os resultados obtidos da comparação da imagem ideal (“*AlgorithmA*”) com as restantes imagens encontram-se na tabela da Figura 4-20.

DataSetProc Name	sensitivity	specificity	Kappa Value
AlgorithmA	Ideal Case	Ideal Case	Ideal Case
AlgorithmB	0.791	0.992	0.643
AlgorithmC	0.528	0.997	0.642
AlgorithmD	0.704	0.992	0.641
AlgorithmE	0.661	0.990	0.639
AlgorithmF	0.688	0.992	0.641
AlgorithmG	0.570	0.997	0.643
AlgorithmH	0.629	0.995	0.642

Figura 4-20: Tabela de resultados da comparação das imagens.

Observando as imagens na Figura 4-19 e os resultados obtidos na Figura 4-20, seria de esperar que os valores de *Sensibilidade* fossem mais próximos de 1 (100%) do que os valores obtidos, pois a *olho nu* todas as imagens dos *DataSet Processados* se parecem com a imagem considerada ideal. Fazendo-se *Zoom* como se pode ver na figura 4-21, existem realmente diferenças como se pode verificar através dos valores da *Sensibilidade*.



Figura 4-21: Zoom visível das quatro primeiras imagens dos *DataSet Processado* escolhidos.

Observando os valores de *Especificidade* todos se encontram acima dos 0.9. O que neste caso era esperado pelo fato do número de *píxeis* definidos pela *máscara* ser muito superior ao número de *píxeis* detetados em cada imagem. A razão da conclusão da *especificidade* provém da equação 4.1 sendo o “*TotaldePixéis*” o número total de *píxeis* de interesse definidos pela *máscara*, caso se considere este um valor muito superior aos restantes, isto é a condição 4.2 então poder-se-á fazer uma aproximação (4.3). Por esse motivo em alguns casos a *Especificidade* não retorna valores de interesse sendo este um desses casos.

$$SP = \frac{VN}{VN+FP} = \frac{TotaldePixeis-FP-FN-VP}{TotaldePixeis-FN-TP} \quad (4.1)$$

$$TotaldePixeis \gg (FP + FN + VP) \quad (4.2)$$

$$SP \approx \frac{TotaldePixeis}{TotaldePixeis} = 1 \quad (4.3)$$

Nesta situação é usado o terceiro valor *Kappa* como valor de interesse. Que é uma relação entre a *Especificidade* e a *Sensibilidade* obtendo-se assim pelo menos mais um valor relevante, que neste caso foi sempre de 0.64 para todas as comparações (Tabela 7).

Kappa de *Cohen* mede a concordância entre dois avaliadores neste caso como se referiu é a *Especificidade* e a *Sensibilidade*. Tendo neste caso o valor de 0.64 significa uma concordância substancial entre os dois [20].

Tabela 4-1: Coeficiente de interpretação Cohen’s kappa

Valor de k	Intrepretação
k<0	Não existe Concordância
0.000≤k≤0.200	Concordância ligeira
0.210≤k≤0.400	Concordância razoável
0.410≤k≤0.600	Concordância moderada
0.610≤k≤0.800	Concordância Substancial
0.810≤k≤1.000	Concordância quase perfeita

A segunda imagem de retina escolhida para testar e obter resultados foi a décima primeira imagem da lista na Figura 4-18 que tem o nome de “*Trial 001 (19)*”. Escolheu-se como *DataSet Processado* ideal “*AlgorithmB*” e escolhendo os restantes *DataSet Processados* visíveis na Figura 4-22, obtêm-se os seguintes resultados da tabela na Figura 4-23.

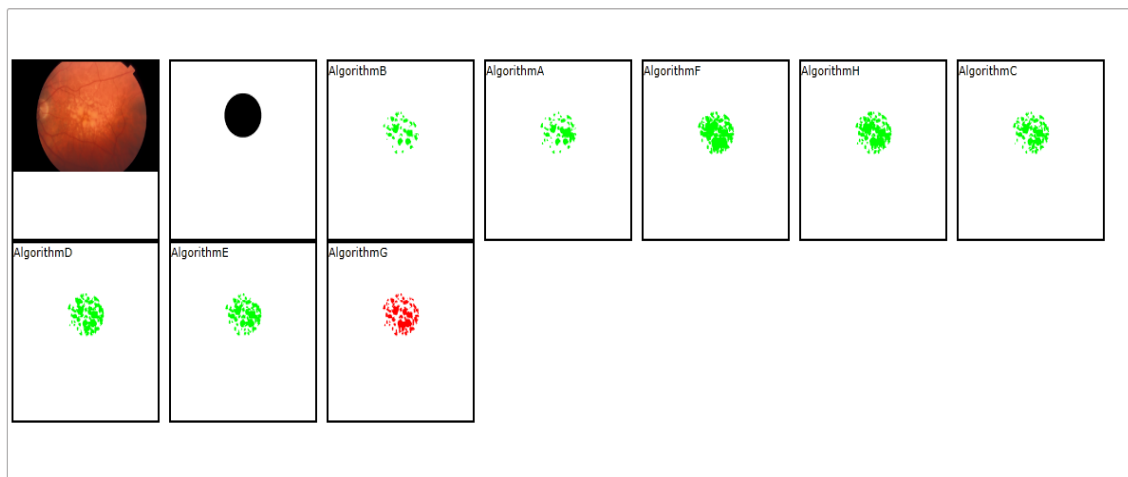


Figura 4-22: Imagem “*Trial 001 (19)*”, a máscara e as imagens dos *DataSet Processados* associadas, sendo “*AlgorithmB*” o *DataSet Processado* considerado ideal.

DataSetProcName	sensitivity	specificity	KappaValue
AlgorithmB	Ideal Case	Ideal Case	Ideal Case
AlgorithmA	0.862	0.876	0.668
AlgorithmF	0.999	0.493	0.474
AlgorithmH	0.996	0.589	0.529
AlgorithmC	0.966	0.725	0.602
AlgorithmD	0.995	0.653	0.566
AlgorithmE	0.981	0.722	0.603
AlgorithmG	0.955	0.764	0.622

Figura 4-23: Resultados da imagem “*Trial 001 (19)*” e o *DataSet Processado* “*AlgorithmB*” escolhido como caso ideal.

Como se pode observar dos resultados, os valores de *sensibilidade* são quase todos acima dos 0.95, exceto no caso do *DataSet Processado* “*AlgorithmA*” que se encontra abaixo dos 0.9.

Isso deve-se ao fato de o número de *píxeis* detetados nas restantes imagens ser um valor significativamente maior do que o da imagem considerada ideal, resultando por esse motivo um maior número de *píxeis* compatíveis na sua comparação.

No caso do *DataSet Processado* “*AlgorithmA*” embora o número de *píxeis* detetados seja também superior em relação ao *DataSet Processado* “*AlgorithmB*” essa diferença não é tão significativa, sendo por isso o *DataSet Processado* com menor valor de *sensibilidade* pois é o que tem o menor número de *píxeis* em comum.

O valor de *sensibilidade* do *DataSet Processado* “*AlgorithmF*” é quase de 100%, a sua razão deve-se à grande diferença entre o número de *píxeis* detetados na imagem do *DataSet Processado* “*AlgorithmF*” e na imagem ideal. Pois o *DataSet Processado* “*AlgorithmF*” detetou quase tantos *píxeis*, como os definidos pela *máscara* como *píxeis* de interesse.

Por esse motivo pode ver-se que o valor da *especificidade* se encontra abaixo dos 0.5 e aqui entra a importância da *especificidade*. Embora o valor da *sensibilidade* seja $0.999 \approx 1$ ou seja de quase 100% de compatibilidade com a imagem ideal, esta erro na detecção incorreta de um grande número de *píxeis* isto é Falsos Positivos. Com estes dois valores (100% da *Sensibilidade* e 50% da *especificidade*) pode chegar-se a uma melhor conclusão da eficácia do algoritmo usado no *DataSet Processado* “*AlgorithmF*”, considerando que o algoritmo usado no *DataSet Processado* “*AlgorithmB*” foi o algoritmo ideal. A conclusão é que o algoritmo do *DataSet Processado* “*AlgorithmF*” não seria um bom algoritmo para se escolher.

Embora tenha detetado corretamente todos os *píxeis* que deveria detetar, também detetou um grande número de *píxeis* que não deveria ter detetado, sendo por esse motivo não recomendado neste caso o algoritmo de “*AlgorithmF*”.

Mantendo-se na mesma imagem “*Trial 001 (19)*” alterou-se só o *DataSet Processado* considerado ideal sendo neste caso o *DataSet Processado* chamado “*AlgorithmF*”, adicionando para comparação todos os restantes *DataSet Processados* existentes que se encontram na Figura 4-24 obtendo os resultados visíveis na Figura 4-25.

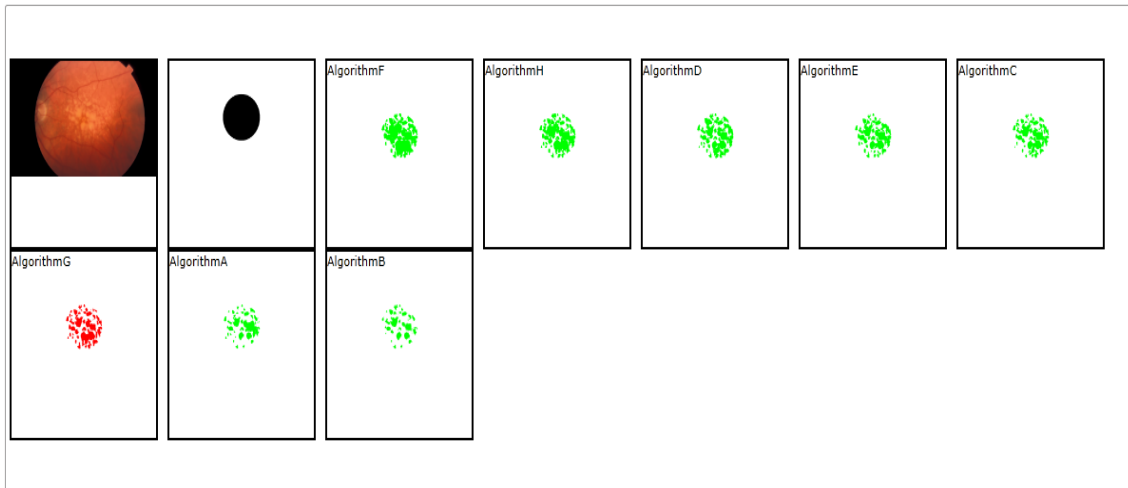


Figura 4-24: Imagem "Trial 001 (19)", a máscara e as imagens dos *DataSet Processados* associadas, sendo "AlgorithmF" o *DataSet Processado* considerado ideal.

DataSetProcName	sensitivity	specificity	KappaValue
AlgorithmF	Ideal Case	Ideal Case	Ideal Case
AlgorithmH	0.826	0.900	0.657
AlgorithmD	0.755	0.909	0.624
AlgorithmE	0.706	0.975	0.620
AlgorithmC	0.698	0.977	0.617
AlgorithmG	0.647	0.976	0.590
AlgorithmA	0.467	0.957	0.495
AlgorithmB	0.399	0.999	0.474

Figura 4-25: Resultados da imagem "Trial 001 (19)" e o *DataSet Processado* "AlgorithmF" escolhido como caso ideal.

Nota-se que com a mudança do *DataSet Processado* considerado ideal, os valores da *Sensibilidade* e da *Especificidade* alteraram significativamente relativamente aos valores anteriormente obtidos na Figura 4-23.

Os valores da *Sensibilidade* diminuíram bastante, nalguns casos abaixo dos 0.5. A quantidade de *píxeis* em comum entre a imagem ideal e as restantes imagens diminuiu consideravelmente em relação ao número total de *píxeis* detetados na imagem ideal.

Os valores da *Especificidade* são maiores ou iguais a 0.9. Isto deve-se ao fato de o número de píxeis erradamente detetados ser muito reduzido, em relação ao total de píxeis definidos pela *máscara* como píxeis de interesse.

A conclusão neste caso é que o algoritmo de *DataSet Processado* “*AlgorithmB*” é um algoritmo não recomendado. Pois foi detetado um número baixo de píxeis do total de píxeis que deveria ter sido detetado, embora em contrapartida tenha detetado também um número baixo de Falso Positivos, de lembrar que o *DataSet Processado* considerado ideal foi o *DataSet Processado* “*AlgorithmF*”.

5 Conclusões e Trabalho Futuro

A plataforma apresentada permitirá ajudar investigadores a testar algoritmos de processamento digital de imagens que tenham desenvolvido para verificar se são eficazes. Tentou-se que a interação do utilizador com a plataforma fosse o mais intuitivo possível, incentivando o utilizador a retornar. Como foi apresentado, já existem algumas plataformas semelhantes, no entanto a diferença é que muitas dessas plataformas são muito restritas ao nível de partilha de imagens e na visualização destas, e são pouco intuitivas e de difícil inserção de novos *DataSet* para comparação.

Outra vantagem da plataforma desenvolvida é que poderá existir comunicação entre os utilizadores através da escrita de comentários, podendo fazer questões sobre como melhorar os algoritmos. Foi disponibilizado uma funcionalidade de armazenamento dos algoritmos usados para geração dos *DataSet Processados*, para que qualquer utilizador possa fazer o *download* dos mesmos, caso estes tenham sido submetidos.

O desenvolvimento da *plataforma online* recorrendo ao uso de *Silverlight* foi gratificante por se terem adquirido novos conhecimentos, especialmente na área de *ASP.NET* através do uso dos *Web Services* e *Generic Handler*.

Silverlight como qualquer tecnologia tem os seus pontos fortes e fracos. Algumas das limitações que se identificaram foi a impossibilidade de importar algumas bibliotecas externas para o projeto. Isto deve-se ao facto de *Silverlight* correr sobre uma versão parcial da *framework .NET* de modo a ocupar o menor espaço possível na sua instalação. Ocorreu esse problema com a biblioteca *OpenCV* (biblioteca de processamento de imagem), sendo necessário implementar as funções de processamento de imagem.

Como trabalhos futuros para melhorar a plataforma seriam, tornar ainda mais intuitivo o seu uso, conseguir diminuir o tempo de compressão e de descompressão das imagens na criação e visualização de um Tópico e ainda implementar outros métodos de comparação de imagens, aumentando assim o leque de indicadores para avaliação da eficácia dos algoritmos.

6 Bibliografia

- [1] R. C. Gonzalez e R. E. Woods, Digital Image Processing Third Edition, Pearson, 2008.
- [2] CSIRO, 14 Outubro 2011. [Online].
- [3] I. S. Institute, 2014. [Online]. Available:
<http://www.isi.uu.nl/Research/Databases/DRIVE/browser.php>.
[Acedido em 23 9 2014].
- [4] Bisque, 2014. [Online]. Available:
http://bisque.ece.ucsb.edu/client_service/. [Acedido em 2014 9 23].
- [5] A. Mora, P. Vieira, A. Manivannan e J. Fonseca, "Automated drusen detection in retinal images using analytical modelling algorithms," vol. 10, n^o 1, 2010.
- [6] R. I. Bank, 2012. [Online]. Available: <http://imagebank.asrs.org/>. [Acedido em 23 9 2014].
- [7] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever e B. v. Ginneken, "Ridge based vessel segmentation in color images of the retina," IEEE Transactions on Medical Imaging, 2004.
- [8] K. Kvilekval, D. Fedorov, B. Obara e A. S. a. B. Manjunath, "Bisque: A Platform for Bioimage Analysis and Management," Bioinformatics, pp. 544-552, Feb 2010.
- [9] P. I. Databases, 2007. [Online]. Available:
<http://www.via.cornell.edu/databases/>. [Acedido em 23 9 2014].
- [10] H. Loureiro, C# 5 com Visual Studio 2012, FCA, 2012.
- [11] A. Carvalho, Práticas de C# Programação Orientada por Objetos, FCA, 2011.
- [12] P. Brown, Silverlight 5 IN ACTION, MANNING, 2012.

- [13] S. Haridas, 2010. [Online]. Available: <http://www.codeproject.com/Articles/99143/BackgroundWorker-Class-Sample-for-Beginners>. [Acedido em 23 9 2014].
- [14] shailesh, "stackoverflow," 2010. [Online]. Available: <http://stackoverflow.com/questions/2448472/what-is-the-difference-between-wcf-and-asmx-web-services>. [Acedido em 23 9 2014].
- [15] B. D. Botella, P. Núñez e R. S. R. , Mastering LOB Development for Silverlight 5: A Case Study in Action, PACKT, 2012.
- [16] E. Lawrence, 2012. [Online]. Available: <http://www.telerik.com/fiddler>. [Acedido em 23 9 2014].
- [17] Silberschatz, Korth e Sudarshan, DATABASE SYSTEM CONCEPTS FIFTH EDITION, McGraw-Hill, 2009.
- [18] S. Kuo, 2008. [Online]. Available: <http://stackoverflow.com/questions/163434/are-nulls-in-a-relational-database-okay>. [Acedido em 23 9 2014].
- [19] J. F.HUGHES e A. V. DAM, COMPUTER GRAPHICS Principles and Practice THIRD EDITIO, Addison wesley, 2013.
- [20] J. R. a. G. G. K. Landis, "The Measurement of Observer Agreement for Categorical Data," Biometrics, n° Biometric Society, 159-74, 1977.
- [21] Y. V. Patil, 2010. [Online]. Available: <http://www.codeproject.com/Articles/139787/What-s-the-Difference-between-WCF-and-Web-Services>. [Acedido em 23 9 2014].