

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**DERMANOSTIC: UM MÉTODO PARA CALCULAR O**  
**DIÂMETRO E EVOLUÇÃO DE LESÕES CUTÂNEAS**

**FELIPE DARUGNA FEY**

**BLUMENAU**  
**2016**

**FELIPE DARUGNA FEY**

**DERMANOSTIC: UM MÉTODO PARA CALCULAR O  
DIÂMETRO E EVOLUÇÃO DE LESÕES CUTÂNEAS**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Aurélio Faustino Hoppe - Orientador

**BLUMENAU  
2016**

# **DERMANOSTIC: UM MÉTODO PARA CALCULAR O DIÂMETRO E EVOLUÇÃO DE LESÕES CUTÂNEAS**

Por

**FELIPE DARUGNA FEY**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Matheus Luan Krueger, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, Mestre – FURB

Blumenau, 6 de dezembro de 2016

## RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo de dispositivos móveis na plataforma Android com um algoritmo de processamento de imagem, tendo o intuito de realizar o cálculo de tamanho e de acompanhar as evoluções de lesões cutâneas de pele ao longo do tempo. O algoritmo inicialmente localiza o gabarito, segmenta a lesão e posteriormente, calcula a proporcionalidade de um em relação ao outro, gerando suas medidas de *pixels* para milímetros. Em seu desenvolvimento, foram utilizadas técnicas de processamento de imagens tais como segmentação, limiarização e detecção de contornos. O aplicativo realiza o cadastro das lesões, podendo acompanhar suas análises e também é possível gerar um gráfico contendo a variação de tamanho da lesão sobre as análises efetuadas. Nos resultados de processamento das imagens obteve-se um acerto de 95,8% na detecção do gabarito e da lesão, 75% para os cálculos do perímetro e de comprimento da lesão e 42% para a área da mesma.

Palavras-chave: Melanoma. Evolução. Processamento de imagens.

## **ABSTRACT**

This work introduces the development of an mobile application in Android platform with an algorithm of image processing, having the objective to perform the calculation of size and following the evolutions of skin moles over the time. The algorithm initially locates the template, segments the mole and then calculates a proportionality of one against the other, generating its measurements from pixels to millimeters. In this development was used image processing techniques such as segmentation, thresholding and contour detection. The application performs the registry of the lesions, being able to follow the analyzes and also it is possible to generate a graph that contains a variation of size on the made analyzes. The results of image processing yield a 95.8% accuracy in template and mole detection, 75% for perimeter and mole length calculations, and 42% for an area of the mole.

Keywords: Melanoma. Evolution. Image processing.

## LISTA DE FIGURAS

Figura 1 - Regra ABCDE (assimetria) .....	14
Figura 2 - Regra ABCDE (borda).....	14
Figura 3 - Regra ABCDE (cor).....	14
Figura 4 - Regra ABCDE (diâmetro) .....	14
Figura 5 - Regra ABCDE (evolução) .....	15
Figura 6 - Exemplo de utilização do filtro de Canny .....	16
Figura 7 - Exemplo de utilização de erosão .....	17
Figura 8 - Exemplo de utilização de dilatação .....	18
Figura 9 - Limiarização simples .....	19
Figura 10 - Tela principal do “Mole Mapper” .....	20
Figura 11 - Tela principal da aplicação .....	21
Figura 12 - Corpus mapping.....	22
Figura 13 - Diagrama de casos de uso .....	25
Figura 14 - Diagrama de atividade .....	26
Figura 15 - Fluxo geral do processamento de imagem.....	28
Figura 16 - Corte de imagem.....	28
Figura 17 - Imagem convertida para escala de cinza.....	29
Figura 18 - Imagem binarizada.....	30
Figura 19 - Técnica de borrão.....	30
Figura 20 - Tentativa por Canny e Threshold.....	31
Figura 21 - Determinando tamanho da linha do gabarito .....	33
Figura 22 - Imagem binarizada.....	34
Figura 23 - Processo de erosão e dilatação .....	35
Figura 24 - Contorno da segmentação da lesão .....	36
Figura 25 - Delimitador das extremidades da lesão .....	37
Figura 26 - Resultado da operação de cálculo .....	38
Figura 27 - Tela principal da aplicação .....	39
Figura 28 - Criação do gabarito .....	39
Figura 29 - Cadastro de pintas .....	40
Figura 30 - Cadastro da lesão .....	40
Figura 31 - Segmentando a lesão.....	41

## LISTA DE QUADROS

Quadro 1 - Comparativo entre os trabalhos correlatos .....	23
Quadro 2 - Código responsável pela conversão para escala de cinza .....	29
Quadro 3 - Código responsável pela binarização da imagem .....	30
Quadro 4 - Borrar a imagem.....	30
Quadro 5 - Binarizando a imagem por Threshold .....	31
Quadro 6 - Encontrando contornos da imagem .....	31
Quadro 7 - Procedimento para encontrar o contorno em forma de quadrado .....	32
Quadro 8 - Medição do gabarito.....	32
Quadro 9 - Aresta do gabarito .....	33
Quadro 10 - Binarização da imagem .....	34
Quadro 11 - Código de dilatação e erosão.....	35
Quadro 12 - Método para encontrar os contornos da lesão .....	35
Quadro 13 - Encontrar elemento com maior circularidade .....	36
Quadro 14 - Cálculo do tamanho da lesão.....	37
Quadro 15 - Cálculo da lesão .....	37
Quadro 16 - Caso de uso Carregar imagem.....	50
Quadro 17 - Caso de uso Segmentar lesão .....	50
Quadro 18 - Caso de uso calcular tamanho da lesão .....	50
Quadro 19 - Caso de uso segmentar gabarito .....	51
Quadro 20 - Resultado das análises realizadas.....	53

## **LISTA DE TABELAS**

Tabela 1 - Resultado dos dados obtidos no comprimento da lesão .....	42
Tabela 2 - Dados obtidos em relação a área da lesão .....	43
Tabela 3 - Dados obtidos em relação ao perímetro da lesão .....	43



## **LISTA DE ABREVIATURAS E SIGLAS**

API – Application Programming Interface

EA – Enterprise Architect

PCA – Principal Component Analysis

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

UML – Unified Modeling Language

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>11</b>
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	12
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>13</b>
2.1 MELANOMA.....	13
2.2 DETECÇÃO DE BORDAS .....	15
2.2.1 Filtro de Canny.....	15
2.3 OPERADORES MORFOLÓGICOS .....	16
2.3.1 Erosão.....	16
2.3.2 Dilatação .....	17
2.4 SEGMENTAÇÃO DE IMAGENS .....	18
2.4.1 Limiarização.....	18
2.5 TRABALHOS CORRELATOS.....	19
2.5.1 Automated analysis of melanocytic lesions .....	19
2.5.2 Melanoma Early Detection using Dual Classifier.....	21
2.5.3 Corpus Mapping: Um aplicativo para o mapeamento corporal de lesões cutâneas .....	22
2.5.4 Comparativo entre os trabalhos correlatos.....	23
<b>3 DESENVOLVIMENTO.....</b>	<b>24</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	24
3.2 ESPECIFICAÇÃO .....	24
3.2.1 Diagrama de casos de uso .....	24
3.2.2 Diagrama de atividades .....	25
3.3 IMPLEMENTAÇÃO .....	27
3.3.1 Técnicas e ferramentas utilizadas.....	27
3.3.2 Operacionalidade da implementação .....	38
3.4 RESULTADOS E DISCUSSÕES.....	42
3.4.1 Aplicação dos testes .....	42
3.4.2 Resultados obtidos nos testes .....	42
<b>4 CONCLUSÕES.....</b>	<b>45</b>
4.1 EXTENSÕES .....	45
<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO.....</b>	<b>50</b>

<b>APÊNDICE B – RESULTADOS OBTIDOS .....</b>	<b>51</b>
--	-----------

## 1 INTRODUÇÃO

Em 2012 foi estimado que cerca de um milhão de pessoas estivessem vivendo com melanoma nos Estados Unidos (SURVEILLANCE, EPIDEMIOLOGY AND RESULT PROGRAM, 2015). A estimativa de novos casos de melanoma nos Estados Unidos para o ano de 2016 é de 76 mil e o número de mortes de 10 mil (SKIN CANCER FOUNDATION, 2016).

Já no Brasil, segundo o Instituto Nacional do Câncer (2016), a estimativa para 2016 é de 5.670 novos casos, dos quais 1.547 representam o número de mortes.

Conforme Peruch (2015, p. 3, tradução nossa):

Melanoma representa menos de 2% de todos os casos de câncer de pele do mundo, mas causa a grande maioria das mortes, além disso, sua incidência está crescendo rapidamente em todo o mundo. O diagnóstico precoce é essencial para aumentar a taxa de sobrevivência, uma vez que a excisão do melanoma for precoce, oferece possibilidade de redução da chance de mortalidade.

Para a detecção do melanoma, normalmente é utilizada uma regra chamada de ABCDE: alterações como assimetria (A), bordas irregulares (B), cor variável da mesma lesão (C), diâmetro maior que 6 mm (D), evolução com aumento da lesão ou alteração das características descritas (E) (MEIRA et al., 2015). No que se refere à realização do diagnóstico, a evolução (E) da lesão é um dos principais fatores. Onde, em uma lesão que teve sua coloração ou tamanho alterado, deve-se obter muita atenção a mesma (MELANOMA RESEARCH FOUNDATION, 2015).

Conforme Araujo (2010), o alto índice de lesões de pele é adquirido de várias formas, e as consequências que elas trazem para o paciente podem vir a se tornar um câncer de pele, tornando sua detecção precoce de extrema importância para definição do tratamento mais adequado. O fato de acompanhar o tamanho de lesões cutâneas benignas a fim de identificar a evolução é um trabalho de devida importância.

Para que a lesão seja medida, geralmente é necessário um equipamento específico, e, como a lesão muitas vezes é curva, essa medição pode não ser totalmente correta, tornando muito complicado comparar sua evolução, bem como verificar o tamanho da mesma, principalmente se esse procedimento for feito por pessoas sem experiência.

O processamento digital de imagens é uma forma confiável de análise, que possibilita analisar, comparar e interpretar os dados através de métodos automatizados. O acompanhamento de lesões de pele via processamento digital de imagens permite avaliar o que num primeiro momento era uma lesão avaliada como lesão benigna possa evoluir rapidamente para melanoma (PAROLIN; HERZER; JUNG, 2010).

Com base nesse cenário, este trabalho apresenta a implementação de um método para calcular o diâmetro (D) e a evolução (E) de lesões cutâneas.

## 1.1 OBJETIVOS

O objetivo geral deste trabalho é criar um método para calcular o tamanho e acompanhar a evolução de lesões cutâneas utilizando técnicas de processamento de imagens.

Os objetivos específicos do trabalho proposto são:

- a) identificar a lesão cutânea e o gabarito a partir de imagens capturadas por um dispositivo móvel;
- b) estabelecer a proporcionalidade entre o gabarito e a lesão cutânea;
- c) apontar se houve crescimento da lesão ao longo do tempo.

## 1.2 ESTRUTURA

O trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta os objetivos e a motivação para o desenvolvimento do trabalho. O segundo trata da fundamentação teórica, explicando sobre conceitos e técnicas desenvolvidas para o trabalho. No terceiro capítulo é descrita a arquitetura do trabalho através dos diagramas, o detalhamento da implementação e os resultados obtidos. No quarto e último capítulo são apresentadas as conclusões e limitações do trabalho, além de sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Essa seção demonstra alguns dos principais conceitos que foram utilizados no trabalho. A seção 2.1 apresenta o conceito de melanoma e demonstra alguns dados a respeito de câncer de pele. A seção 2.2 descreve técnicas de detecção de bordas em processamento de imagens. A seção 2.3 explica sobre operadores morfológicos e sua utilização prática. A seção 2.4 aborda segmentação de imagens, como reconhecer objetos de interesse em uma imagem. Por fim, a seção 2.5 demonstra os trabalhos correlatos ao aplicativo desenvolvido.

### 2.1 MELANOMA

Segundo o Instituto Nacional do Câncer (2016), o melanoma cutâneo é um tipo de câncer de pele que tem origem nos melanócitos (células produtoras de melanina, substância que determina a cor da pele) e tem predominância em adultos brancos. A doença se manifesta através do surgimento de uma pinta escura de bordas irregulares, na maioria dos casos não apresenta sintomas, mas o tumor pode vir acompanhado de coceira e descamação. Em caso de lesão pigmentada pré-existente, o tumor irá evoluir apresentando mudanças de cores que podem variar do marrom claro ao preto, aumento da superfície e das bordas da lesão (SOARES, 2008).

Uma técnica utilizada pelos dermatologistas para diagnosticar o câncer de pele, e consequentemente o melanoma, é a dermatoscopia. Trata-se de um exame não invasivo, realizado *in vivo*, que analisa superfícies pigmentadas na pele humana. O aparelho mais utilizado pelos médicos é o dermatoscópio, que consiste em um aparelho portátil ótico que permite ampliar a lesão de 10 a 100 vezes (ALENCAR, 2015). Na análise de lesões, além da dermatoscopia, se destaca a utilização da regra ABCDE.

A regra ABCDE é um guia para diagnóstico visual das lesões a fim de detectar um possível melanoma. Essa regra foi introduzida em 1985 como acrônimo ABCD e em 2004 expandida para ABCDE (JENSEN; ELEWSKI, 2015).

Ela classifica a lesão por:

- a) assimetria: se uma linha for traçada no meio da lesão, significa que ela é simétrica, ou seja, os dois lados vão corresponder. A assimetria significa que ambos os lados não correspondem ao traçar essa linha. Um exemplo de assimetria pode ser visto na Figura 1;

Figura 1 - Regra ABCDE (assimetria)



Fonte: Skin Cancer Foundation (2015).

- b) borda: as bordas da lesão tendem a ser irregulares, conforme mostra a Figura 2;

Figura 2 - Regra ABCDE (borda)



Fonte: Skin Cancer Foundation (2015).

- c) cor: a maioria das lesões benignas são de uma cor só. Quando tendo uma variedade de cores tende a ser um fator de alerta. Um número de diferentes tons de marrom, bege ou preto pode aparecer. O melanoma pode também tornar-se vermelho, branco ou azul, como representa a Figura 3;

Figura 3 - Regra ABCDE (cor)



Fonte: Skin Cancer Foundation (2015).

- d) diâmetro: lesões benignas tendem a ter um diâmetro menor que as malignas. Melanomas geralmente possuem um diâmetro maior que 6 mm, mas podem ser menores quando detectados pela primeira vez, conforme demonstra a Figura 4;

Figura 4 - Regra ABCDE (diâmetro)



Fonte: Skin Cancer Foundation (2015).

- e) evolução: lesões benignas tem a mesma aparência ao longo do tempo. Quando uma lesão muda sua forma, tamanho ou cor pode ser sinal de melanoma, como mostra a Figura 5.

Figura 5 - Regra ABCDE (evolução)



Fonte: Skin Cancer Foundation (2015).

Segundo Melanoma Research Foundation (2015), a evolução (E) da lesão se tornou o fator mais importante a se considerar em um diagnóstico. Thomas et al. (1998) concluíram que a evolução é o critério mais específico entre as características da regra ABCDEs. Em um estudo de características dermatoscópicas realizado com 169 lesões pigmentadas descobriu-se que lesões observadas por dermatologistas para serem ao mesmo tempo não uniforme (isto é, a partilha de alguns dos critérios ABC), e que tiveram alterações, tinham pelo menos quatro vezes mais chance de ser melanoma do que as lesões que não apresentavam estas características (ABASSI, 2004).

## 2.2 DETECÇÃO DE BORDAS

Segundo Boaventura (2010, p. 76), bordas e contornos são características úteis, uma vez que representam uma imagem pelo limite dos objetos e separação das regiões não similares em termos de intensidade de *pixels*, além disso, apresentam informação essencial de um objeto de interesse na imagem.

Conforme Nunes e Conci (2007, p. 185), o processo de detecção de bordas consiste na verificação da variação dos valores de luminosidade de uma imagem visando localizar e realçar os *pixels* de borda, por meio da ampliação do contraste entre esta e o fundo. A seção 2.2.1 descreve sobre o método Canny utilizado para a detecção de bordas.

### 2.2.1 Filtro de Canny

O filtro de Canny é um detector de bordas otimizado e foi criado por John Canny em 1983 e publicado em 1986, com o objetivo inicial de resolver o problema de sensibilidade a ruídos/texturas na detecção de bordas (PIRES, 2008).

Ainda conforme Pires (2008, p. 15):

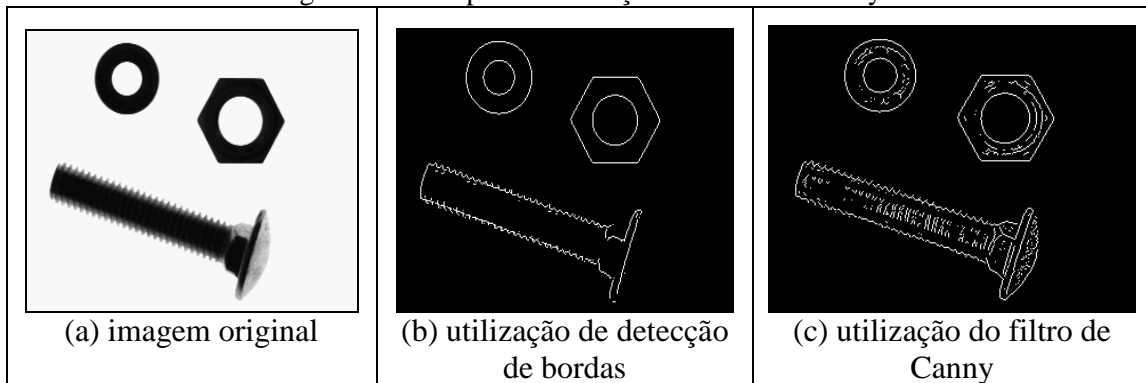
O filtro de Canny deve satisfazer três critérios básicos de desempenho: boa detecção, boa localização e boa resposta. Em outras palavras, um detector de bordas deve ser capaz de: minimizar o número de falsos positivos (*pixels* detectados, mas que não são bordas) e falsos negativos (*pixels* de bordas que não são detectados); minimizar a distancia entre a borda detectada e a borda verdadeira e; minimizar a probabilidade de múltipla detecção de uma borda (isto é, o algoritmo deve fornecer uma resposta única para cada *pixel* de borda na imagem).



Para processar as bordas são seguidos os seguintes passos: suavização de uma imagem por um filtro gaussiano, processar o gradiente da matriz suavizada, a magnitude e orientação do gradiente são computadas por fórmulas de conversão de coordenadas retangulares e utilizada à técnica de supressão não máxima (VALE; POZ, 2002).

Um exemplo de utilização pratica do filtro de Canny pode ser visto na Figura 6.

Figura 6 - Exemplo de utilização do filtro de Canny



Fonte: elaborado pelo autor.

No exemplo demonstrado acima, a Figura 6a representa a imagem original, onde então é realizado um filtro de detecção de bordas padrão, resultando na Figura 6b. Já a Figura 6c demonstra a utilização do filtro de Canny sobre a imagem original, podendo verificar os detalhes obtidos visivelmente na detecção.

## 2.3 OPERADORES MORFOLÓGICOS

A morfologia matemática é uma ferramenta que auxilia a extração de componentes úteis na imagem para a descrição de forma de uma região (GONZALES; WOODS, 2000, p. 369). Conforme Costa e Cesar Jr. (2000, p. 255), pré-processamento de imagens, filtros de ruídos, detecção de formas e associação de padrões são alguns dos problemas onde a morfologia matemática vem sendo aplicada.

Segundo Facon (1996, p. 2), o princípio básico da morfologia matemática consiste em extrair uma informação relativa à geometria e à topologia de um conjunto desconhecido de uma imagem pela transformação a partir de outro conjunto completamente definido chamado EE (Elemento estruturante). Ainda conforme Facon (1996), existem dois operadores morfológicos básicos (dilatação e erosão).

### 2.3.1 Erosão

Conforme Queiroz e Gomes (2001, p. 22), o objetivo da erosão é remover os *pixels* que não atendem a um dado padrão. Matematicamente, pode-se definir a erosão de uma

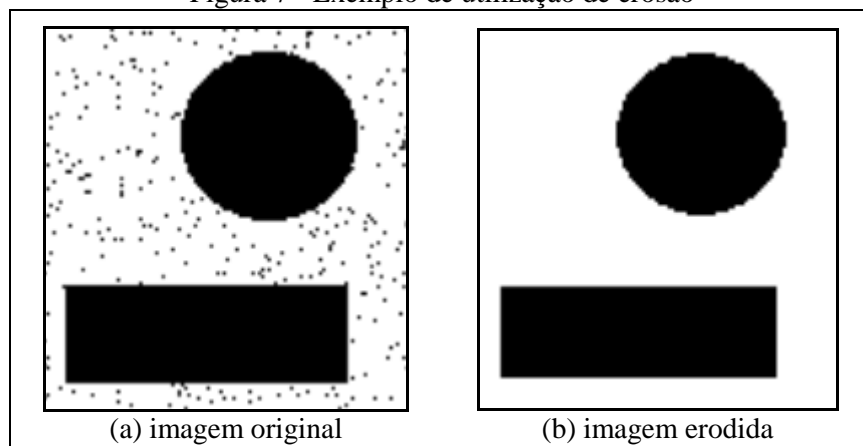
imagem  $f$  pelo elemento estruturante  $B$  da seguinte maneira:  $\varepsilon_B(f) = \{x \in E : B_x \subset f\}$  onde  $B_x$  representa o elemento estruturante  $B$  transladado na posição  $x$ .

Facon (2011, p. 64) diz que:

Segundo essa definição deve-se deslizar o elemento estruturante  $B$  sobre a imagem  $f$  e para cada pixel  $x$  verificar a configuração de sua vizinhança em relação à estrutura do elemento estruturante  $B$ . Por ser binários, a imagem  $f$  e o elemento estruturante  $B$  contém dois tipos de informação, o fundo e os *pixels* relevantes. O significado dessa definição é que o elemento estruturante  $B_x$ , posicionado e centrado no pixel  $x$  de  $f$ , tenta aparelhar-se com a vizinhança de  $x$ . Entende-se que cada pixel relevante de  $B_x$  deve encontrar-se na mesma posição na vizinhança de  $x$ . Caso seja verificado, o pixel  $x$  na imagem erodida será considerado um pixel relevante e será preservado. Caso contrário, ele será considerado como irrelevante e será apagado.

Um exemplo de erosão pode ser verificado na Figura 7.

Figura 7 - Exemplo de utilização de erosão



Fonte: Facon (2011).

Na Figura 7a pode se verificar que existem ruídos em preto, ao aplicar a técnica de erosão nos conjuntos de preto da imagem pode ser verificado que na Figura 7b o ruído desaparece.

### 2.3.2 Dilatação

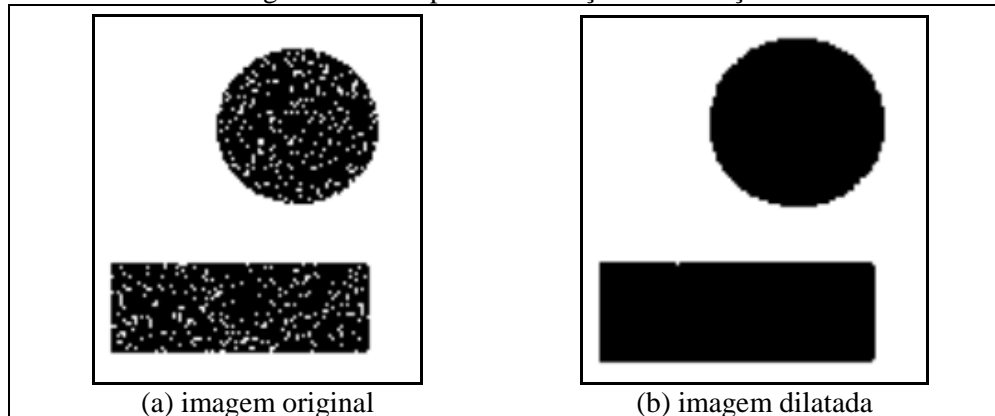
Segundo Queiroz e Gomes (2001, p. 22), a dilatação partir da qual uma pequena área relacionada a um pixel é alterada para um dado padrão. Matematicamente, pode-se definir dilatação de uma imagem  $f$  pelo elemento estruturante  $B$  como:  $\delta_B(f) = \{x \in f : B_x \cap f\}$  onde  $B_x$  representa o elemento estruturante  $B$  transladado na posição  $x$  (FACON, 2011).

Ainda conforme Facon (2011, p. 66):

Segundo essa equação, o elemento estruturante  $B_x$ , posicionado e centrado em cada pixel  $x$  de  $f$  desliza na imagem  $f$  e verifica uma possível interseção com a vizinhança de  $x$ . Caso seja verdadeiro, o ponto central na imagem resultado será considerado um *pixel* relevante e será marcado como tal. Caso contrário, ele será considerado como irrelevante e será apagado.

Um exemplo de dilatação pode ser verificado na Figura 8.

Figura 8 - Exemplo de utilização de dilatação



Fonte: Facon (2011).

Na Figura 8a pode ser verificado que existe um conjunto de ruídos branco internamente nos elementos em preto, após ser aplicado o método de dilatação esses ruídos desaparecem e os elementos que estão em preto são preenchidos como mostra a Figura 8b.

## 2.4 SEGMENTAÇÃO DE IMAGENS

A segmentação de imagens é um importante método utilizado em problemas de análise de imagens. Suas aplicações variam desde o controle de qualidade industrial à medicina, navegação de robôs, aplicações militares, exploração geofísica, sensoriamento remoto, etc (CUFÍ et al., 2002).

Segundo Negri e Silva (2012), o objetivo da segmentação é particionar a imagem em um conjunto de regiões visualmente distintas e uniformes com relação a alguma propriedade, como o nível de cinza, textura ou cor.

Conforme Faria (2010), os algoritmos clássicos utilizados na segmentação de imagem são baseados em uma das duas propriedades básicas dos valores de intensidade, descontinuidade e similaridade. No primeiro caso, a segmentação é baseada em mudanças bruscas nos valores de intensidade dos *pixels* da imagem. No segundo caso, a segmentação de imagem é baseada na sua partição em regiões semelhantes de acordo com um conjunto de critérios predefinidos.

### 2.4.1 Limiarização

A limiarização é uma das abordagens mais importantes de segmentação de imagens. O objetivo dela é separar a imagem entre duas regiões, o fundo e o objeto de interesse, as demais

regiões são classificadas como não interessantes. Nesta prática, obtém-se uma imagem com o objeto preto sobre o fundo branco (ABOUD NETA et al., 2008).

A limiarização utiliza uma imagem de entrada  $f(x, y)$  de  $N$  níveis de cinza que produz uma saída uma imagem  $g(x, y)$ , chamada de imagem limiarizada, cujo número de níveis de cinza é menor que  $N$ . Normalmente,  $g(x, y)$  apresenta 2 (dois) níveis de cinza, conforme ilustrado na Figura 9.

Figura 9 - Limiarização simples

$$g(x, y) = \begin{cases} \text{objeto} & \text{se } f(x, y) > T \\ \text{fundo} & \text{se } f(x, y) \leq T \end{cases}$$

Fonte: Aboud Neta et al. (2008).

Onde, os *pixels* rotulados com 1 correspondem aos objetos e os *pixels* rotulados com 0 correspondem ao fundo e  $T$  é um valor de tom de cinza predefinido denominado limiar (FEITOSA, 2015).

A técnica de limiarização de histograma baseia-se nos picos e vales do histograma de uma imagem para separar os seus objetos do plano de fundo. Em imagens monocromáticas, os picos e vales podem ser facilmente identificados a partir do histograma de luminosidade da imagem. No caso de imagens coloridas, é necessária a análise dos histogramas das três componentes de cores da imagem (MAIA, 2014).

## 2.5 TRABALHOS CORRELATOS

Foram encontrados alguns trabalhos correlatos, sendo que a maioria geralmente tem por definição apenas a segmentação e remoção de fatores que atrapalham o resultado final. Na questão “evolução”, foi encontrado apenas o trabalho de Peruch (2015).

Dentre os trabalhos selecionados estão: Automated analysis of melanocytic lesions (PERUCH, 2015); Melanoma Early Detection using Dual Classifier (CHAITANYA; PAVANI, 2015); Corpus Mapping: um aplicativo para o mapeamento corporal de lesões cutâneas (LIMA, 2015).

### 2.5.1 Automated analysis of melanocytic lesions

Peruch (2015) criou uma solução chamada de Mole Mapper para ajudar dermatologistas durante o processo de diagnóstico. Essa solução implementa 3 algoritmos principais:

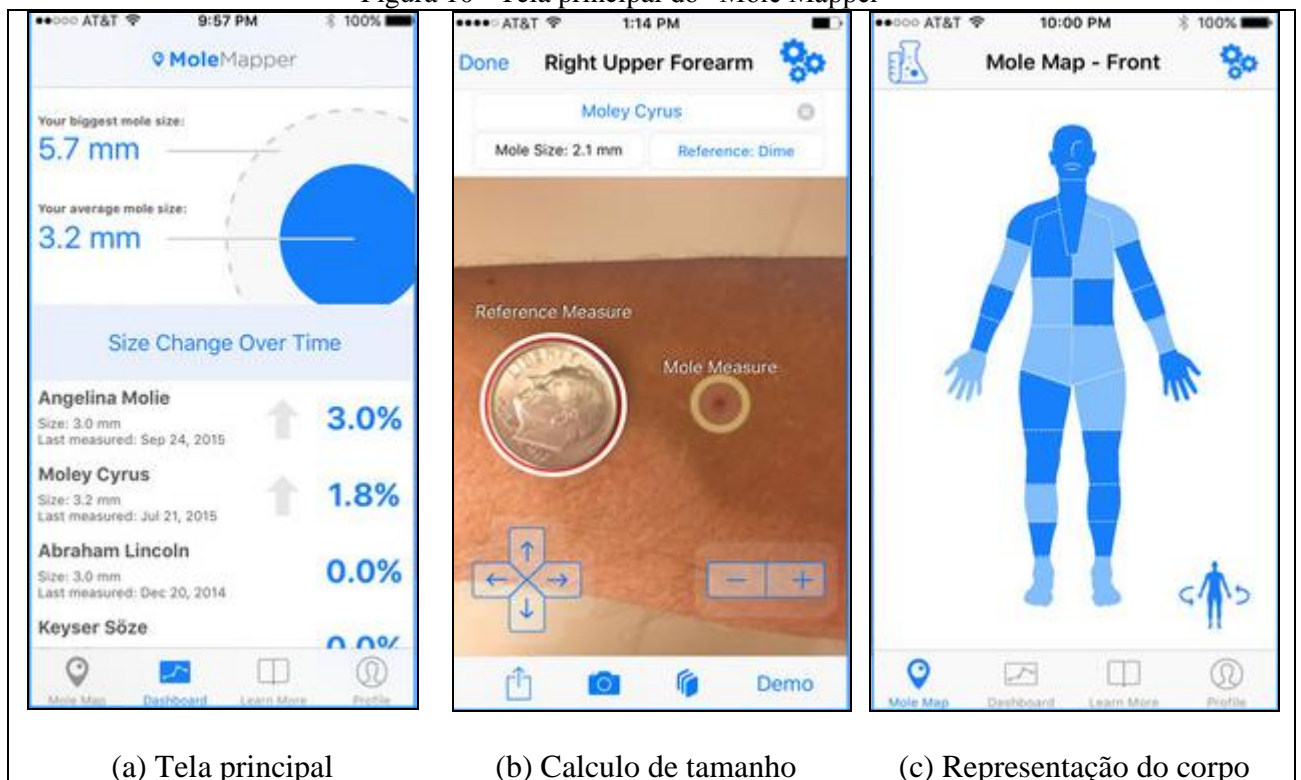
- a) um algoritmo de segmentação de lesão melanocítica, que permite a extração de

informações úteis sobre a forma da mesma;

- b) um algoritmo de “barbear virtual”, o que aumenta a precisão e a robustez dos outros algoritmos e permite ao dermatologista uma lesão sem pelos;
- c) um algoritmo de registro da evolução da lesão ao longo do tempo destacando anomalias e variações inesperadas.

A Figura 10a demonstra a tela principal do aplicativo, onde é possível identificar também o aplicativo demonstrando a evolução da lesão em questão, na Figura 10b pode-se observar a lesão sendo medida e na Figura 10c a representação do corpo humano permitindo selecionar uma parte e poder cadastrar lesões para a mesma.

Figura 10 - Tela principal do “Mole Mapper”



Fonte: elaborado pelo autor.

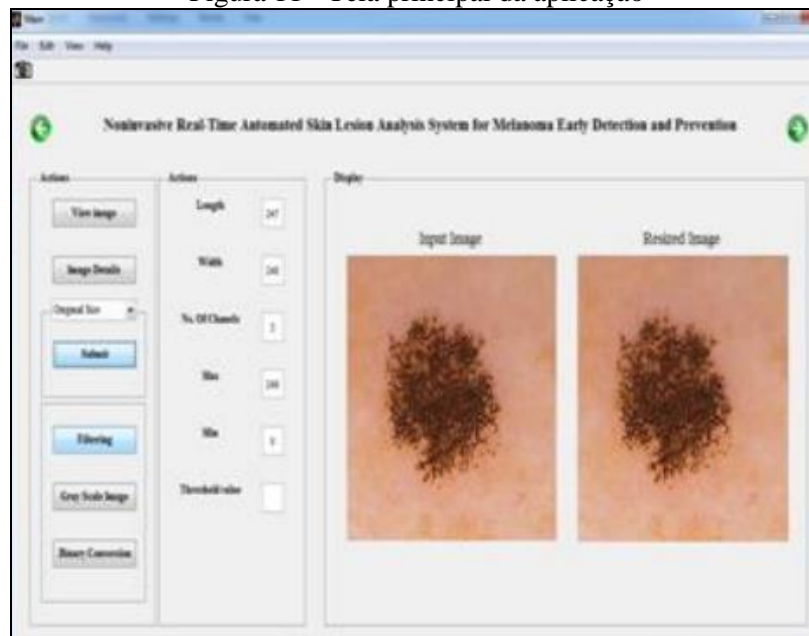
Para a parte de segmentação têm-se cinco estágios, dentre eles, o pré-processamento que consta com o barbear virtual, iluminação da imagem e balanceamento de cor. A segunda fase é o Principal Component Analysis (PCA) que é uma ferramenta padrão para análise estatística das observações em um espaço multidimensional. Dentro do PCA feito o agrupamento das cores. Após finalizar a segunda fase, o próximo estágio é a redução de ruído. A quarta fase é o agrupamento de cor, que associa a cada cor com o número de *pixels* da mesma cor, separando cor e, portanto *pixels* em dois grupos, correspondendo respectivamente à lesão e a pele não lesionada. Esta fase é a parte principal da técnica e que imita o processo

cognitivo de dermatologistas. Por último, o pós-processamento onde ocorre o mapeamento da segmentação para voltar à cor original e classifica-lo em lesional ou não-lesional.

### 2.5.2 Melanoma Early Detection using Dual Classifier

O trabalho de Chaitanya e Pavani (2015) teve por objetivo criar um aplicativo que simula um smartphone e em tempo real faz a detecção do câncer de pele. O aplicativo possui os seguintes módulos: aquisição da imagem, detecção e remoção de pelos, segmentação da lesão, extração das características, classificação e análise de performance. A Figura 11 apresenta a tela principal do protótipo.

Figura 11 - Tela principal da aplicação



Fonte: Chaitanya e Pavani (2015).

O sistema usa uma base de dados do hospital Pedro Hispano Hospital de Portugal para fins de desenvolvimento e testes. O banco de dados contém um total de 200 imagens de lesões, incluindo benigno, atípico e casos de melanoma.

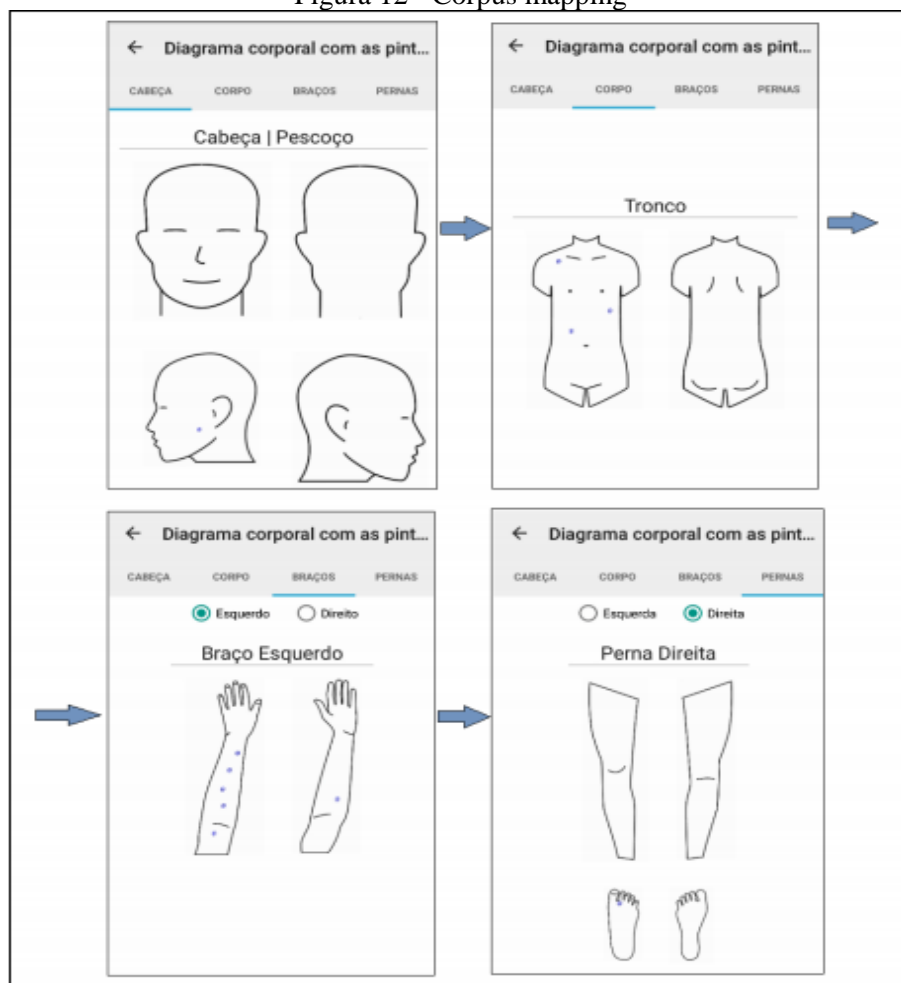
A segmentação da imagem é uma parte fundamental para o processo, onde ocorre o particionamento da imagem em vários segmentos (conjunto de *pixels*). Cada um dos *pixels* em uma região é semelhante a alguma característica ou propriedade calculada, como por exemplo, a intensidade da cor e a textura. Nesta fase, é utilizado o algoritmo de interpolação como o *Marching Cubes*, que reconstitui a segmentação para reconstruções em 3D.

O protótipo desenvolvido classifica a lesão como benigno, atípico ou melanoma, e um alerta é enviado ao usuário para que este seja consultado com um médico. Os resultados obtidos tiveram uma alta taxa de acertos.

### 2.5.3 Corpus Mapping: Um aplicativo para o mapeamento corporal de lesões cutâneas

Lima (2015) teve como objetivo desenvolver um aplicativo para a plataforma Android que realizasse o mapeamento corporal de lesões cutâneas, técnica utilizada na detecção precoce do melanoma. Neste trabalho foi utilizada a linguagem de programação Java e a ferramenta Android Studio, além da API do Google Drive para efetuar o *backup* dos dados. A Figura 12 mostra a tela principal do Corpus mapping.

Figura 12 - Corpus mapping



Fonte: Lima (2015).

Foi utilizado um gabarito para estimar o tamanho da pinta na captura da imagem, o próprio aplicativo instrui o usuário como criar o gabarito. Após a criação do gabarito é feita a captura da imagem, onde envolve a seleção da região de interesse por parte do aplicativo. Em seguida, é necessário vincular a imagem capturada com uma região corporal. Por fim, pode-se visualizar as imagens, no qual é selecionada a parte do corpo para verificar o histórico das pintas, inserir anotação e associar uma nova imagem com o local da pinta sendo visualizada.

A aplicação teve um resultado satisfatório, apesar de que alguns usuários que testaram a aplicação informaram que precisavam de alguém para utilizar a câmera. Esta sugestão foi

motivada pela dificuldade do usuário segurar o gabarito e o smartphone para capturar a imagem. Em comparativo com as outras aplicações o aplicativo consta com *backup* de dados, padronização da captura, possibilidade de cadastrar mais de um paciente, visualizar duas imagens de lesões simultaneamente, podendo ser utilizado pelos próprios pacientes.

#### 2.5.4 Comparativo entre os trabalhos correlatos

O Quadro 1 apresenta um comparativo entre as principais características dos trabalhos correlatos.

Quadro 1 - Comparativo entre os trabalhos correlatos

Características	Trabalhos	Peruch (2015)	Chaitanya e Pavani (2015)	Lima (2015)
Plataforma		IOS	Desktop	Android
Cálculo de tamanho da lesão		Sim	Não	Não
Segmentação de lesão		Sim	Sim	Não
Mapeamento das lesões		Sim	Não	Sim
Classificação da lesão em lesional ou não lesional		Sim	Sim	Não

Fonte: elaborado pelo autor.

A partir do Quadro 1 pode-se observar que o trabalho realizado por Peruch (2015) é o mais completo contemplando todas as características da regra ABCDE. Porém, só está disponível para a plataforma IOS. Lima (2015) tinha por objetivo realizar o mapeamento das lesões. Enquanto Chaitanya e Pavani (2015) tinham como intuito identificar qual era a possibilidade da lesão ser um câncer de pele.

Relacionando as características com o trabalho proposto, a questão de cálculo de tamanho é realizada apenas por Peruch (2015), onde este calcula o comprimento da lesão utilizando como gabarito uma moeda para realizar proporcionalização.

Em relação à segmentação da lesão, apenas o trabalho de Lima (2015) não realiza este procedimento devido ao seu objetivo que era apenas fazer o mapeamento das lesões cutâneas. Já nos trabalhos de Peruch (2015) e Chaitanya e Pavani (2015) este procedimento foi realizado pois é de suma importância para realizar a classificação da lesão em lesional ou não.



### 3 DESENVOLVIMENTO

Neste capítulo são demonstradas as etapas do desenvolvimento do protótipo. Na seção 3.1 são apresentados os requisitos funcionais e não funcionais da aplicação. A seção 3.2 demonstra a especificação do protótipo. A seção 3.3 apresenta de forma explicativa a implementação do algoritmo de detecção de imagem. Por fim, a seção 3.4 apresenta os resultados dos testes, sugestões e melhorias.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O aplicativo desenvolvido possui os seguintes Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

- a) permitir a captura de imagens utilizando a câmera do dispositivo Android (RF);
- b) permitir o cadastro de lesões a partir de partes do corpo do usuário (RF);
- c) realizar a segmentação do gabarito (RF);
- d) realizar a segmentação da lesão (RF);
- e) proporcionalizar o tamanho da lesão a partir do gabarito (RF);
- f) criar um gráfico evolutivo da lesão em questão (RF);
- g) ser desenvolvido para a plataforma Android (RNF);
- h) utilizar a linguagem de programação C# com o plugin Xamarin (RNF);
- i) ser desenvolvido utilizando o *wrapper* EmguCV para o algoritmo de processamento de imagem (RNF).

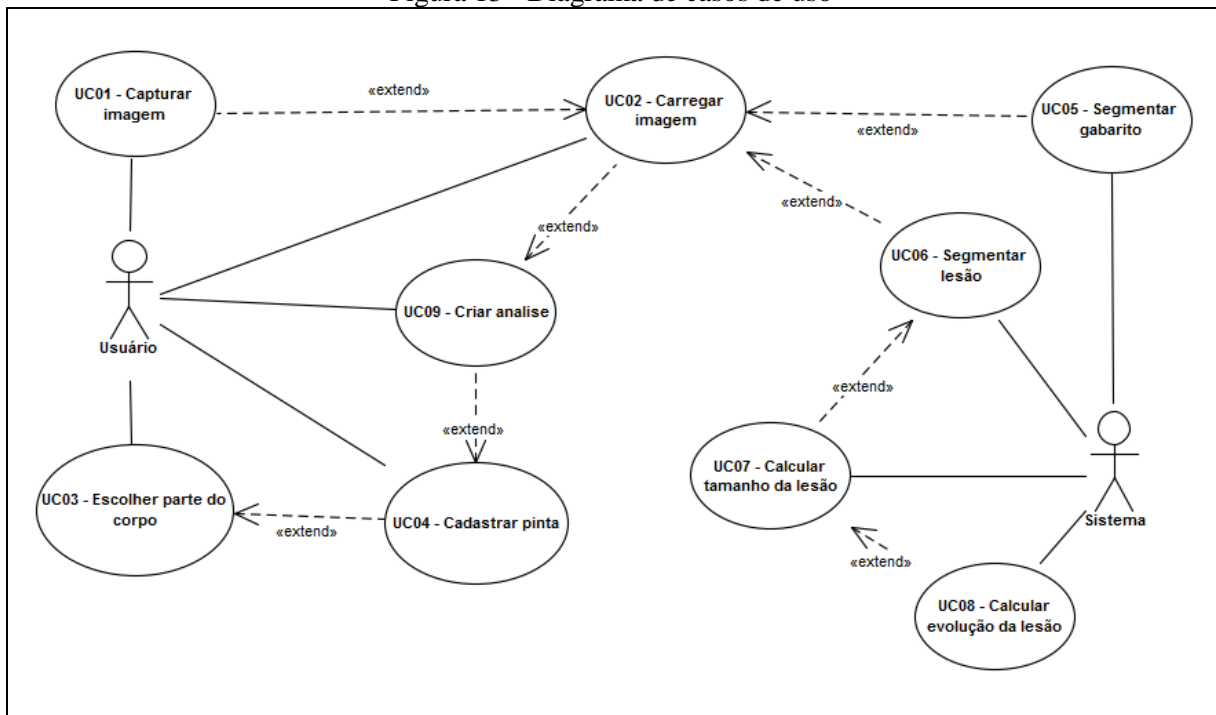
#### 3.2 ESPECIFICAÇÃO

A especificação do protótipo foi representada através da Unified Modeling Language (UML), utilizando a ferramenta Enterprise Architect (EA). Neste trabalho foram desenvolvidos os diagramas de casos de uso e de atividades.

##### 3.2.1 Diagrama de casos de uso

Os casos de uso que representam as funcionalidades da aplicação são exibidos na Figura 13. O detalhamento dos mesmos encontra-se no apêndice A.

Figura 13 - Diagrama de casos de uso



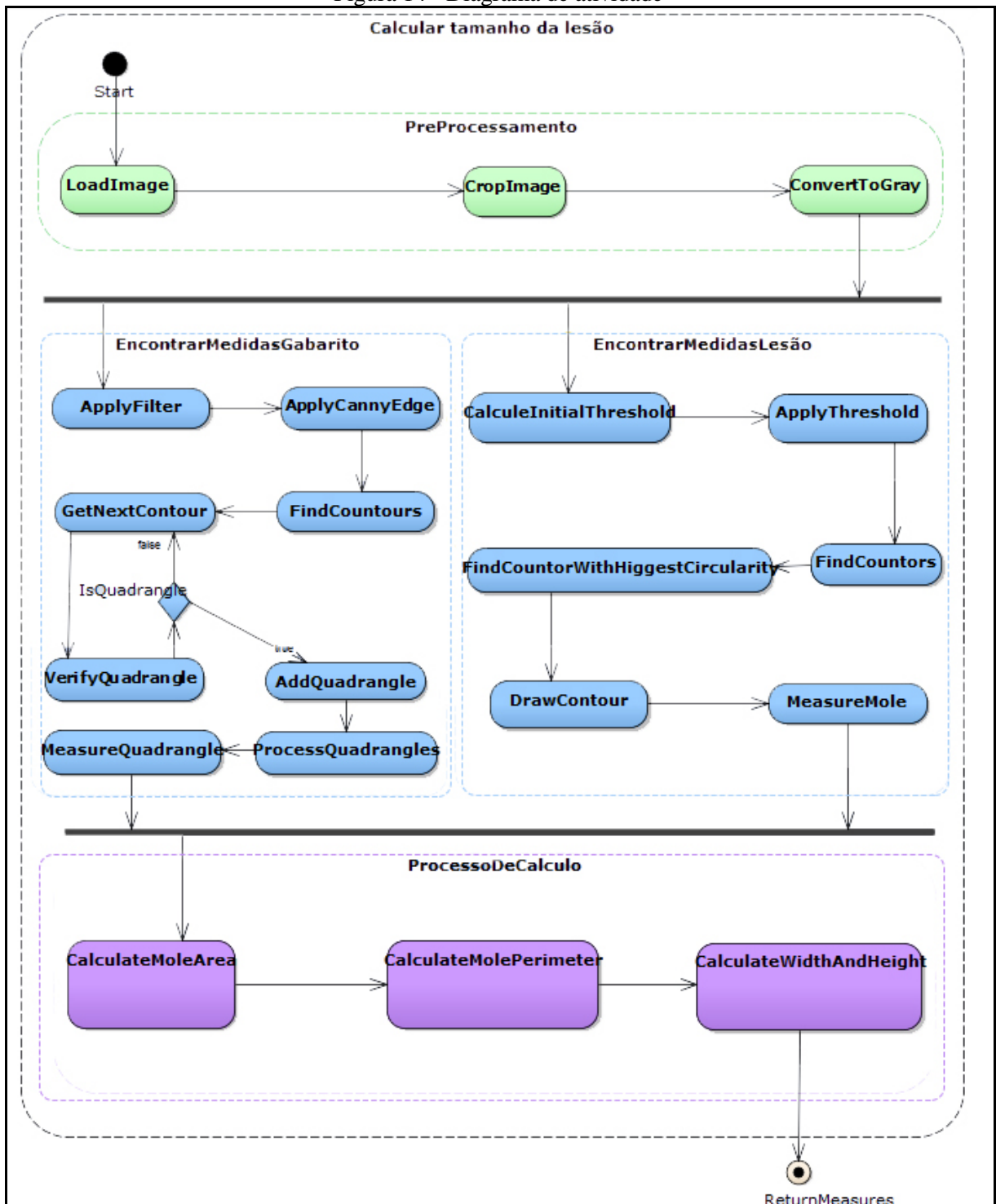
Fonte: elaborado pelo autor.

A partir da Figura 13 percebe-se que a aplicação dispõe do ator Usuário como manipulador do smartphone e o ator Sistema responsável pelo algoritmo de processamento de imagem. O caso de uso UC04 - Cadastrar pinta permite ao usuário cadastrar uma lesão em uma parte do corpo. Tendo a lesão cadastrada, o usuário pode realizar a sua análise a partir do UC09 - Analisar lesão que irá executar indiretamente o UC02 - Carregar imagem ou UC01 - Capturar imagem. A partir do cadastrado da lesão, o sistema irá executar o UC05 - Segmentar gabarito e o UC06 - Segmentar lesão a fim de localizar o gabarito e a lesão dentro da imagem. Identificados tais elementos (gabarito e lesão), o sistema tentará estabelecer a proporcionalidade entre eles por meio do UC07 - Calcular tamanho da lesão.

### 3.2.2 Diagrama de atividades

O diagrama de atividades é responsável por demonstrar o fluxo de execução dos processos necessários para medir a lesão, a Figura 14 demonstra o processo de medição da lesão.

Figura 14 - Diagrama de atividade



Fonte: elaborado pelo autor.

Na Figura 14 pode-se observar que o processo de cálculo se inicia pelo pré-processamento, onde a imagem será carregada e recortada pelo usuário a fim de estabelecer sua área de interesse. A partir disso, a imagem é convertida para a escala de cinza e, posteriormente, o sistema realizará os procedimentos para identificar a lesão e o gabarito.

Para encontrar as medidas do gabarito, inicialmente aplica-se o filtro de *Canny* para encontrar os contornos dos elementos existentes na imagem. A partir disso, é feita uma

varredura para identificar os contornos em formato de quadrado. Dentre os identificados, o elemento que possuir a maior área será considerado o gabarito. Ao qual, terá suas medidas retornadas em *pixels*.

Para encontrar a lesão, primeiramente é realizada uma binarização. Posteriormente, são encontradas todas as componentes conexas da imagem. A partir delas, verifica-se qual possui a maior circularidade, sendo considerada a lesão. Por fim, destaca-se o contorno correspondente para facilitar a visualização da lesão encontrada por parte do usuário e, ao mesmo tempo, calcula-se a sua medida em *pixels*.

Tendo encontrado as medidas do gabarito e da lesão é possível estabelecer a proporcionalidade entre eles. Para isso, são levados em consideração a área, o perímetro e o comprimento de ambos.

### 3.3 IMPLEMENTAÇÃO

Neste capítulo são mostradas as técnicas, ferramentas e a operacionalidade da implementação. A seção 3.3.1 apresenta o detalhamento das ferramentas e as técnicas utilizadas. A seção 3.3.2 demonstra o processo operacional do aplicativo.

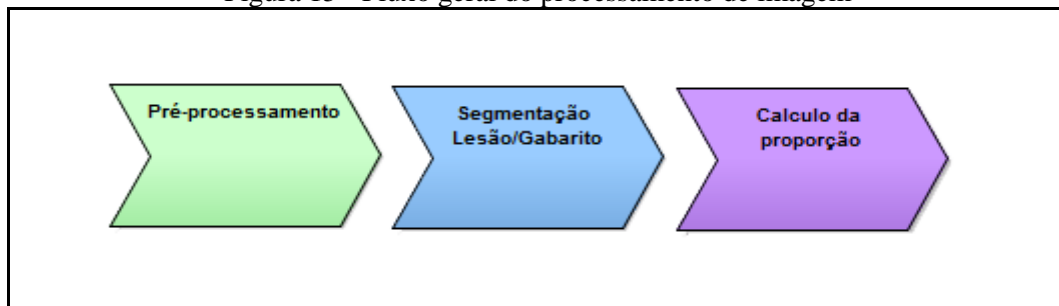
#### 3.3.1 Técnicas e ferramentas utilizadas

A aplicação foi desenvolvida utilizando a linguagem de programação C# na IDE Visual Studio 2015. As tecnologias utilizadas no desenvolvimento estão listadas abaixo:

- a) Xamarin: a plataforma Xamarin permite desenvolvimento de aplicativos Android e iOS utilizando a linguagem C# dentro da IDE Visual Studio;
- b) EmguCV: é um chamado *wrapper* da biblioteca de processamento de imagens OpenCV utilizando a linguagem C#. O EmguCV pode ser compilado no Visual Studio e tem compatibilidade com o Xamarin, podendo assim ser utilizado para desenvolver aplicativos Android;
- c) SQLite (Android): o SQLite é um repositório de dados que simula um banco de dados relacional. A plataforma Android suporta nativamente o SQLite.

Nas próximas seções são detalhados os passos adotados na implementação do aplicativo para medir a evolução de lesões cutâneas. A Figura 15 é parte referente ao processamento de imagens e possui as seguintes etapas: pré-processamento, segmentação, cálculo da proporcionalidade.

Figura 15 - Fluxo geral do processamento de imagem



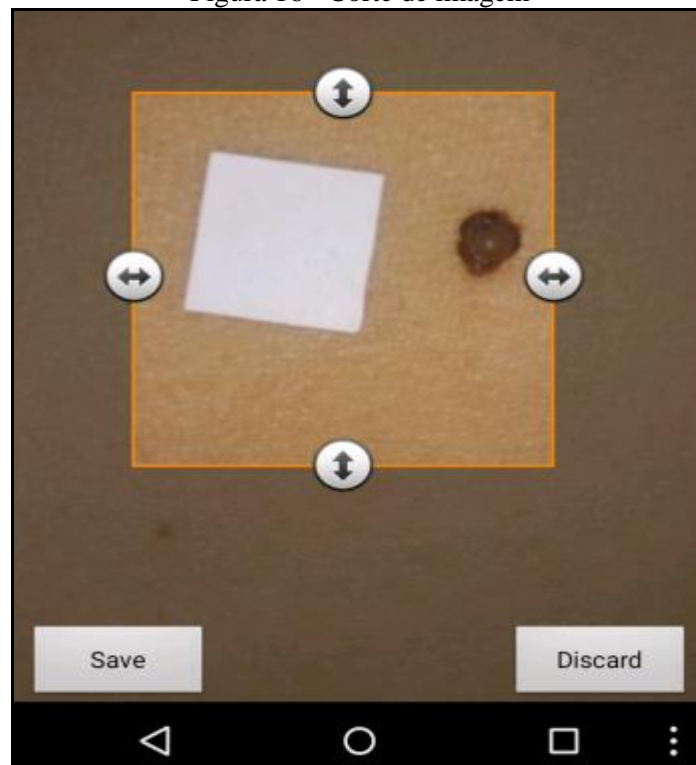
Fonte: elaborado pelo autor.

O detalhamento está dividido em etapas e possui exemplos do código fonte implementado, assim como os resultados dessas implementações. A seção a seguir, descreve a etapa de pré-processamento.

### 3.3.1.1 Pré-processamento

Nesta seção a imagem é carregada a partir do aplicativo, seja da galeria do celular ou por meio de captura da câmera. Tendo a imagem carregada, o usuário pode recortá-la para focalizar apenas nos elementos principais da imagem, ou seja, o gabarito e a lesão. Dessa forma, ele reduzirá a possibilidade de erros no processo de segmentação. A Figura 16 mostra o processo de corte de imagem realizado pelo usuário do aplicativo.

Figura 16 - Corte de imagem



Fonte: elaborado pelo autor.

Depois de realizar o recorte da imagem, ela é convertida para escala de cinza. Para realizar este procedimento foi utilizado o método `CvtColor`, conforme mostra o código do Quadro 2.

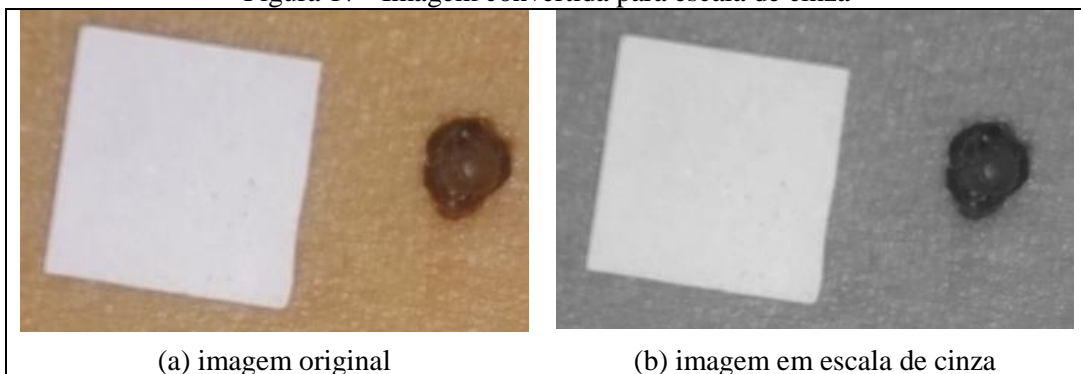
Quadro 2 - Código responsável pela conversão para escala de cinza

```
1. Mat imageGray = new Mat(image.Size, DepthType.Cv8U, 255);
2. CvInvoke.CvtColor(image, imageGray, ColorConversion.Rgba2Gray);
```

Fonte: elaborado pelo autor.

A Figura 17a demonstra a imagem original e a Figura 17b mostra o resultado da operação de conversão para escala de cinza.

Figura 17 - Imagem convertida para escala de cinza



Fonte: elaborado pelo autor.

O próximo passo é a segmentação da imagem. Esta atividade é realizada para separar o plano de fundo da imagem das regiões de interesse, no caso, o gabarito e a lesão.

### 3.3.1.2 Segmentação

Nesta etapa são detalhados os passos para realizar a identificação do gabarito e da lesão cutânea. A seção 3.3.1.2.1 descreve o processo e a identificação do gabarito. Já a seção 3.3.1.2.2 apresenta como foi feito o isolamento da lesão.

#### 3.3.1.2.1 Identificação do Gabarito

Com a imagem pré-processada, o próximo passo é encontrar as medidas do gabarito. Para isso, foi utilizado o filtro `Canny` disponibilizado pela API `EmguCV`. Ele verifica a mudança de intensidade dos *pixels*, destacando os contornos dos objetivos existentes na imagem. Na utilização desse método foram informados dois parâmetros de *threshold*, se um gradiente de pixel é maior que o maior *threshold*, o pixel é considerado contorno. Se o valor for abaixo do *threshold* ele será rejeitado. Se o gradiente de pixel estiver entre os dois *thresholds*, então ele será aceito somente se estiver conectado a um pixel que está acima do maior *threshold*. O Quadro 3 demonstra essa técnica utilizada.

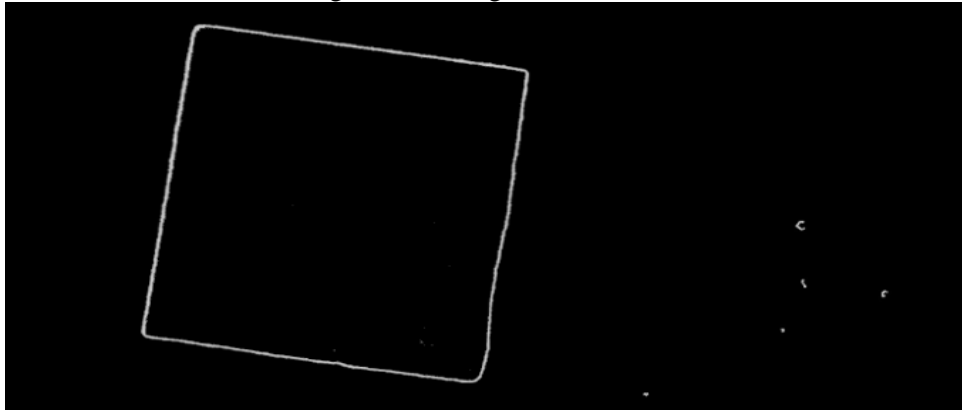
Quadro 3 - Código responsável pela binarização da imagem

```
1. CvInvoke.Canny(imageGray, binaryImg, 200, 90)
```

Fonte: elaborado pelo autor.

Os valores 200 e 90 foram os valores inseridos para encontrar os contornos. O resultado da utilização de *Canny* pode ser visto na Figura 18.

Figura 18 - Imagem binarizada



Fonte: elaborado pelo autor.

Muitas vezes o método de *Canny* não é efetivo, pois alguns contornos não são contínuos. Por causa disso, foram realizados outros procedimentos para encontrar os contornos e consecutivamente o gabarito. Primeiramente, tentou-se reconstruir as áreas faltantes, conforme mostra o código do Quadro 4.

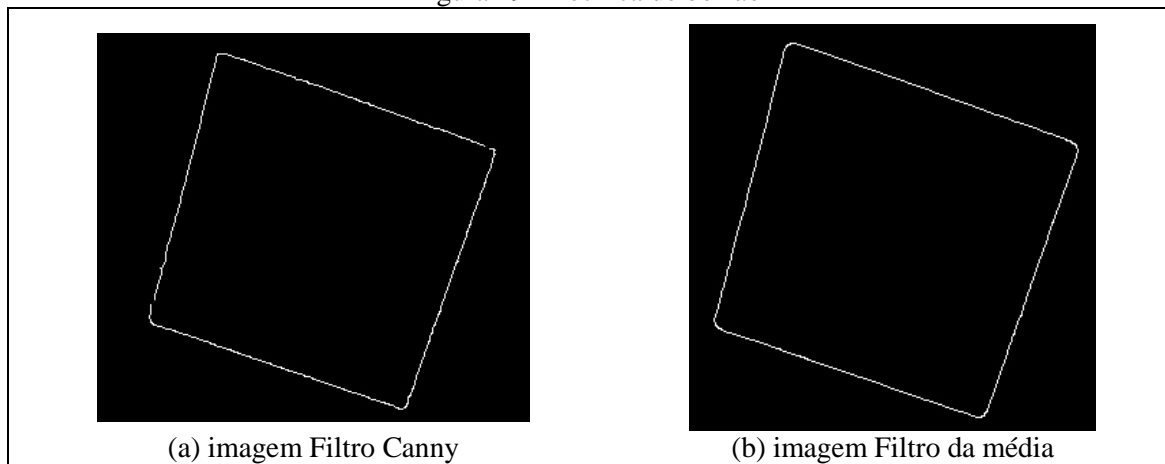
Quadro 4 - Borrar a imagem

```
1. CvInvoke.MedianBlur(imageGray, imageBlurred, 9);
2. CvInvoke.Canny(imageBlurred, imageGrayGray, 200, 90);
```

Fonte: elaborado pelo autor.

Para realizar a reconstrução, foi utilizado o método `MedianBlur` que causa um borrão na imagem e, que no caso em questão, melhorou o alinhamento das linhas que representam o contorno do gabarito como demonstra Figura 19.

Figura 19 - Técnica de borrão



Fonte: elaborado pelo autor.

Caso não seja possível estabelecer o contorno do gabarito a partir do método de detecção de bordas, tentou-se encontrá-lo pela técnica de limiarização, utilizando o método `Threshold` e os valores de 180 e 255, conforme pode ser visto no Quadro 5.

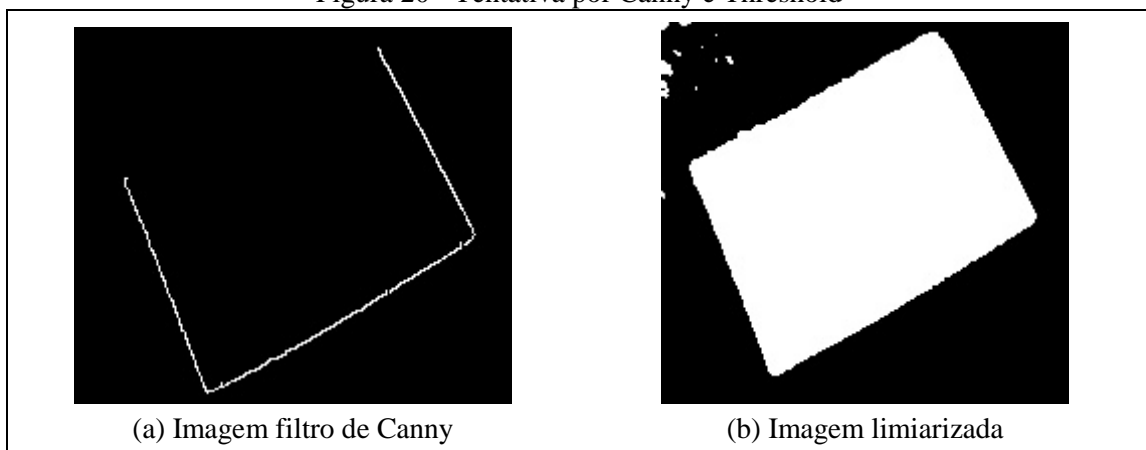
Quadro 5 - Binarizando a imagem por `Threshold`

```
1. CvInvoke.Threshold(imageGray, imageGrayGray, 180, 255,
   ThresholdType.Binary);
```

Fonte: elaborado pelo autor.

A Figura 20a retrata um caso onde o filtro de `Canny` foi aplicado e não gerou bons resultados. Já a Figura 20b demonstra a utilização da técnica de limiarização. A principal diferença entre limiarização e `Canny` é que por limiarização o objeto é identificado como um todo e por `Canny` apenas seus contornos.

Figura 20 - Tentativa por `Canny` e `Threshold`



Fonte: elaborado pelo autor.

A partir da imagem binarizada, pode-se encontrar seus contornos. Para isso, utilizou-se o método `FindContours`, que retorna os contornos da imagem binarizada e os armazena num vetor de vetor de pontos. Este processo é demonstrado no Quadro 6.

Quadro 6 - Encontrando contornos da imagem

```
1. VectorOfVectorOfPoint contours = new VectorOfVectorOfPoint();
2. CvInvoke.FindContours(binaryImg, contours, null, RetrType.External,
   ChainApproxMethod.ChainApproxSimple);
```

Fonte: elaborado pelo autor.

O parâmetro `RetrType.External` indica o tipo de contorno que será retornado. No caso, serão recuperados apenas os contornos mais externos. Ou seja, caso existam contornos internos a um objeto maior, estes serão ignorados, sendo adicionados no vetor os contornos apenas os elementos mais externos.

No passo seguinte, os contornos encontrados foram percorridos no intuito de deixar apenas os que tinham a forma de quadrado. Para isso, foram utilizadas as técnicas para identificar o quadrado, conforme mostra o Quadro 7.



Quadro 7 - Procedimento para encontrar o contorno em forma de quadrado

```

1.  CvInvoke.ApproxPolyDP(contours[i], approx,
    CvInvoke.ArcLength(contours[i], true) * 0.02, true);
2.  if (approx.Size == 4 && CvInvoke.ContourArea(approx) > 400 &&
    CvInvoke.IsContourConvex(approx)) {
3.      double maxCosine = 0;
4.      for (int j = 2; j < 5; j++)
5.      {
6.          double cosine = Angle(approx[j%4], approx[j-2], approx[j-1]);
7.          maxCosine = Math.Max(maxCosine, cosine);
8.      }
9.      if (maxCosine < 0.3)
10.     {
11.         squares.Push(contours[i]);
12.     }
13. }

```

Fonte: elaborado pelo autor.

O método `ApproxPolyDP` aproxima uma curva com outra curva com menos vértices para que a distância entre elas seja menor ou igual a precisão especificada como mostra a linha 1.

Para ser considerado um quadrado, a aproximação deve ter quatro lados, uma área maior que quatrocentos e ser convexo. Para verificar se o contorno é convexo ou não, é utilizado o método `IsContourConvex`, conforme mostra a linha 2.

Caso o contorno passe por todas as verificações serão percorridas as arestas para encontrar o maior cosseno, descrito na linha 7. Se todos os ângulos são pequenos, aproximadamente 90 graus, é um quadrado válido, e este será adicionado no *array* de retorno.

### 3.3.1.2.2 Cálculo do tamanho

Ao encontrar o *array* de contornos do gabarito pode-se realizar o procedimento para estabelecer o tamanho dele. Foram efetuados os cálculos do perímetro, da área e do comprimento.

Para encontrar o perímetro do gabarito utilizou-se o método `ArcLength`, conforme mostra a linha 2. Já a área foi encontrada através do método `ContourArea` exemplificado na linha 3 do Quadro 8.

Quadro 8 - Medição do gabarito

```

1.  private void MeasureQuadrangle(VectorOfVectorOfPoint contours,
    VectorOfPoint quadrangle){
2.      var quadranglePerimeter = CvInvoke.ArcLength(quadrangle, true);
3.      var quadrangleArea = CvInvoke.ContourArea(quadrangle, false);

```

Fonte: elaborado pelo autor.

Muitas vezes os contornos não são quatro linhas exatamente retas e sim vários pontos uniformes. Para encontrar o tamanho do comprimento do gabarito foi necessário utilizar o método `MinAreaRect`, que desenha um retângulo nos pontos extremos do contorno, assim como mostra o Quadro 9.

Quadro 9 - Aresta do gabarito

```

1.  var rotatedRect = CvInvoke.MinAreaRect (quadrangle);
2.  var rect_points = rotatedRect.GetVertices ();
3.  List<LineSegment2D> lines = new List<LineSegment2D> ();
4.  for (int i = 0; i < rect_points.Length; i++)
5.  {
6.  lines.Add(new LineSegment2D(new Point ((int)rect_points[i].X,
    (int)rect_points[i].Y), new Point ((int)rect_points[(i + 1) % 4].X,
    (int)rect_points[(i + 1) % 4].Y)));
7.  }
8.  var quadrangleLength = (float)lines.Max(t => t.Length);

```

Fonte: elaborado pelo autor.

Como resultado, obtém-se os quatro vértices do quadrado no retorno do método `GetVertices`. A partir deles, é possível realizar uma ligação entre os pontos e instanciar um objeto `LineSegment2D` para criar uma linha reta ao qual será possível medi-la. A Figura 21 demonstra a criação do retângulo e os vértices encontrados.

Figura 21 - Determinando tamanho da linha do gabarito



Fonte: elaborado pelo autor.

Com as linhas encontradas, é realizado o procedimento para pegar a maior linha reta. Este procedimento é realizado pois, às vezes, o usuário não consegue recortar um quadrado com todos os seus lados iguais, assim nas próximas análises o lado sempre será o maior.

### 3.3.1.3 Segmentação da lesão

Nesta etapa, após ser encontradas as medidas do gabarito, é realizada a segmentação da lesão, para que posteriormente seja calculado a sua proporcionalidade.

O primeiro passo a ser realizado é a binarização da imagem. Para isso, utiliza-se o processo de limiarização, que separa o objeto que será analisado do fundo da imagem. O valor inicial do *threshold* é definido da seguinte maneira, caso não exista nenhuma análise anterior à realização do exame atual, seu valor é 70, se não, ele utilizará a última amostragem como

valor definido. O valor 70 foi utilizado baseado nos testes realizados. Depois de estabelecido o valor do *threshold*, utilizou-se o método `Threshold` para realizar a binarização, conforme demonstra o Quadro 10.

Quadro 10 - Binarização da imagem

```
1. CvInvoke.Threshold(imgDest, imgDest, _threshold, 255,  
    Emgu.CV.CvEnum.ThresholdType.BinaryInv);
```

Fonte: elaborado pelo autor.

O método utiliza o valor encontrado no parâmetro `_threshold` até o valor de branco (255) para encontrar as áreas de interesse na imagem. O parâmetro `ThresholdType.BinaryInv` informa ao método o tipo de binarização. Neste caso, os pixels maiores que o parâmetro `_threshold` serão convertidos para branco (255), conforme mostra a Figura 22.

Figura 22 - Imagem binarizada



Fonte: elaborado pelo autor.

A partir da Figura 22, pode-se verificar que a imagem contém alguns ruídos, pois existem pixels fora do limiar estabelecido no método `Threshold`. Para minimizar este problema, foram aplicados os operadores morfológicos de dilatação e erosão para preencher os ruídos existentes na imagem. O Quadro 11 demonstra o código utilizado para realizar este procedimento.

Quadro 11 - Código de dilatação e erosão

```

1. var se = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new
   Size(3, 3), new System.Drawing.Point(-1, -1));
2. CvInvoke.Erode(imgDest, imgDest, se, new System.Drawing.Point(-1, -
   1), 1, Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(1));
3. se = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new
   Size(12, 12), new System.Drawing.Point(-1, -1));
4. CvInvoke.Dilate(imgDest, imgDest, se, new System.Drawing.Point(-1, -
   1), 1, Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(1));
5. se = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new
   Size(12, 12), new System.Drawing.Point(-1, -1));
6. CvInvoke.Erode(imgDest, imgDest, se, new System.Drawing.Point(-1, -
   1), 1, Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(1));
7. se = CvInvoke.GetStructuringElement(ElementShape.Rectangle, new
   Size(3, 3), new System.Drawing.Point(-1, -1));
8. CvInvoke.Dilate(imgDest, imgDest, se, new System.Drawing.Point(-1, -
   1), 1, Emgu.CV.CvEnum.BorderType.Default, new MCvScalar(1));

```

Fonte: elaborado pelo autor.

A técnica de erosão tem o objetivo de eliminar pixels dispersos e a dilatação de preencher os espaços vazios aumentando o tamanho das regiões brancas. Na aplicação dessas técnicas foram utilizadas duas máscaras de tamanho 3x3 e 12x12 *pixels*. Os valores foram baseados em diversos testes realizados, onde a imagem não altera seu tamanho. O resultado deste processo é apresentado na Figura 23.

Figura 23 - Processo de erosão e dilatação



Fonte: elaborado pelo autor.

Tendo a imagem sem ruídos, pode-se encontrar o contorno dela. Para isso, foi utilizado o método `FindContours`, conforme demonstra o Quadro 12.

Quadro 12 - Método para encontrar os contornos da lesão

```

1. CvInvoke.FindContours(imgDest, contours, hierarchy,
   Emgu.CV.CvEnum.ReTrType.External,
   Emgu.CV.CvEnum.ChainApproxMethod.ChainApproxSimple);

```

Fonte: elaborado pelo autor.

Os contornos serão armazenados no vetor `contours`, o parâmetro `RetrType.External` indica que só serão considerados os contornos mais externos. O resultado pode ser verificado na Figura 24.

Figura 24 - Contorno da segmentação da lesão



Fonte: elaborado pelo autor.

Em certos casos, alguns contornos não pertinentes à lesão podem ser encontrados. Para desconsiderá-los, foi verificado a sua circularidade, conforme mostra o Quadro 13.

Quadro 13 - Encontrar elemento com maior circularidade

```

1.  double biggerCircularity = 0.00;
2.  int contournIdx = -1;
3.  for (int i = 0; i < contours.Size; i++)
4.  {
5.      var perimeter = CvInvoke.ArcLength(contours[i], false);
6.      var area = CvInvoke.ContourArea(contours[i], false);
7.      var circularity = 4 * Math.PI * (area / Math.Pow(perimeter, 2));
8.      circ.Add(circularity);
9.      circ2.Add(perimeter / quadradoPerimeter);
10.     if(circularity <= 1.00)
11.     {
12.         if(circularity > biggerCircularity && area > 50)
13.         {
14.             biggerCircularity = circularity;
15.             contournIdx = i;
16.         }
17.     }
18. }

```

Fonte: elaborado pelo autor.

A fórmula do cálculo da circularidade se dá por  $R = (4 \cdot \text{PI} \cdot A) / P^2$ , onde  $A$  corresponde à área do elemento e  $P$  o perímetro do mesmo, quanto mais próximo de 1, mais próximo de um círculo é o elemento. Também é feita a verificação do tamanho da área, pois caso seja um contorno muito pequeno a circularidade do mesmo tende a ser alta, porém, o mesmo não corresponde a uma lesão. Tendo encontrado o contorno que representa a lesão, pode-se realizar o cálculo da proporção.

### 3.3.1.3.1 Cálculo do tamanho da lesão

Após ter a lesão devidamente segmentada o próximo passo é encontrar a área, o perímetro e a altura e largura da mesma. Este procedimento é realizado a partir do trecho de código descrito no Quadro 14.

Quadro 14 - Cálculo do tamanho da lesão

```

1.   var rectMole = CvInvoke.BoundingRectangle(contours[contournIdx]);
2.   float moleW = rectMole.Width;
3.   float moleH = rectMole.Height;
4.   var moleArea = CvInvoke.ContourArea(contours[contournIdx]);
5.   var molePerimeter = CvInvoke.ArcLength(contours[contournIdx],
      false);

```

Fonte: elaborado pelo autor.

Para encontrar a área da lesão foi utilizado o método `ContourArea` e para o perímetro o método `ArcLength`. Para realizar o cálculo da altura e largura da lesão foi necessário utilizar o método `BoudingRectangle` que desenha um retângulo ao redor do contorno da lesão possibilitando pegar o valor de suas extremidades, conforme mostram as linhas 1, 2 e 3 do Quadro 14. O resultado pode ser visualizado na Figura 25.

Figura 25 - Delimitador das extremidades da lesão



Fonte: elaborado pelo autor.

Tendo as medidas do gabarito e da lesão, pode-se realizar o cálculo da proporcionalidade do tamanho em *pixels* para medidas reais (centímetros). O Quadro 15 mostra o cálculo realizado para proporcionalizar as medidas dos objetos.

Quadro 15 - Cálculo da lesão

```

1.   float real_w_patch = 10; float real_h_patch = 10;
2.   float quadrangleWH = quadrangleLenght;
3.   var real_w_mole = moleW / quadrangleWH * real_w_patch;
4.   var real_h_mole = moleH / quadrangleWH * real_h_patch;
5.   var real_moleArea = (moleArea / quadrangleArea) * real_w_patch;
6.   var real_molePerimeter = (molePerimeter / quadranglePerimeter) *
      real w patch;

```

Fonte: elaborado pelo autor.

Na linha 1, é informado o valor real da altura e largura do gabarito, no caso 10 milímetros, este que será usado como base da proporcionalização.

A linha 2 refere-se a altura e largura do gabarito, este valor foi obtido em sua segmentação. Nas linhas 3 e 4 realiza-se o cálculo de altura e largura em medidas reais da lesão. Onde, a altura/largura da lesão foi dividida pela altura/largura do gabarito e depois multiplicada pelo seu valor real.

O cálculo para o valor real da área da pinta é feito pela divisão da área da pinta pela área do gabarito, multiplicado pelo valor real do gabarito, conforme mostra a linha 5.

E, por fim, o perímetro é calculado pela divisão do perímetro da pinta pelo perímetro do gabarito, sendo multiplicado pelo valor real do gabarito, como mostra a linha 6.

A Figura 26 demonstra o resultado dessa operação. Ao qual, pode-se verificar que a lesão possui um comprimento de 4 milímetros, uma área de 1,1 centímetros e um perímetro de 2,9 centímetros.

Figura 26 - Resultado da operação de cálculo

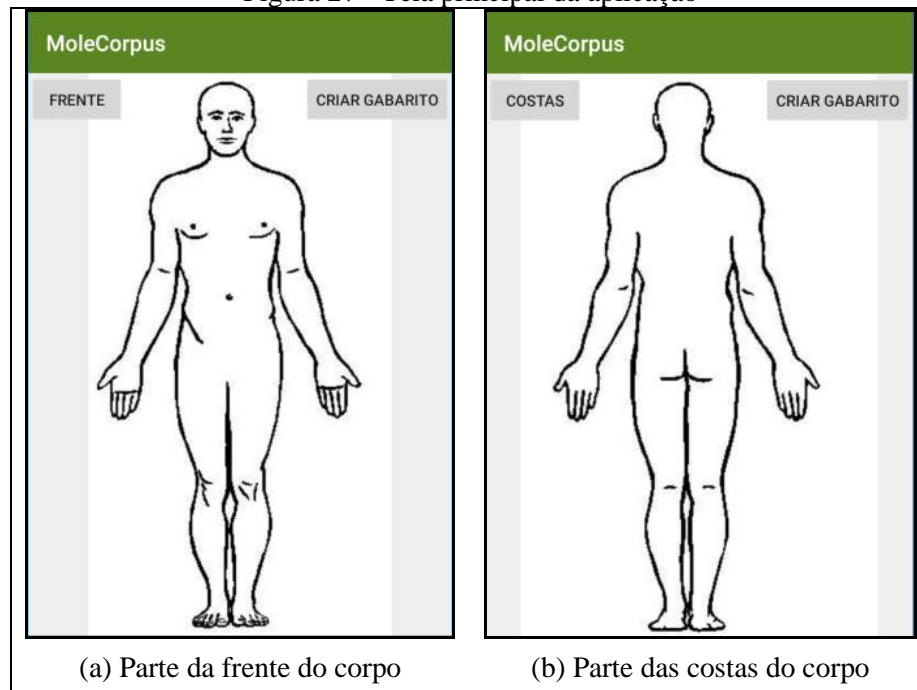


Fonte: elaborado pelo autor.

### 3.3.2 Operacionalidade da implementação

Esta seção é destinada a mostrar o funcionamento da aplicação. A tela inicial mostra uma representação do corpo humano e os botões para Criar gabarito e o botão que indica a localização da pinta (frente ou costas). Ao acionar o botão Frente o corpo será rotacionado para frente, conforme por ser visualizado na Figura 27.

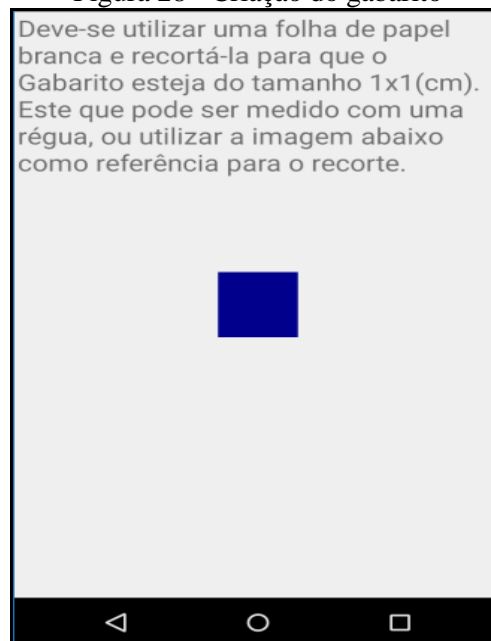
Figura 27 - Tela principal da aplicação



Fonte: elaborado pelo autor.

O botão `Criar gabarito` demonstra ao usuário como realizar a criação do gabarito. Ao qual, será utilizado para medir a lesão. Este procedimento pode ser visto na Figura 28.

Figura 28 - Criação do gabarito

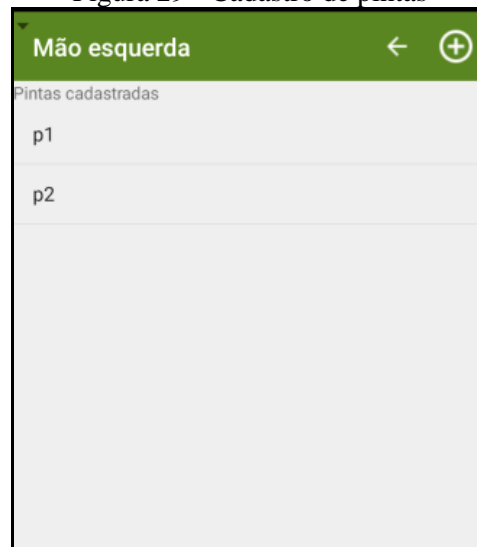


Fonte: elaborado pelo autor.

Ao selecionar uma parte do corpo será aberta a visualização das lesões cadastradas para aquele usuário, conforme mostra a Figura 29.



Figura 29 - Cadastro de pintas



Fonte: elaborado pelo autor.

Ao clicar no botão *Adicionar*, o usuário pode vincular uma nova lesão na parte do corpo escolhido. Ele também pode verificar informações detalhadas da pinta. Ao selecionar uma lesão, uma nova tela é aberta contendo a última amostragem realizada, o tamanho desta análise e a imagem da lesão, assim como, o gráfico evolutivo da lesão como mostra a Figura 30.

Figura 30 - Cadastro da lesão

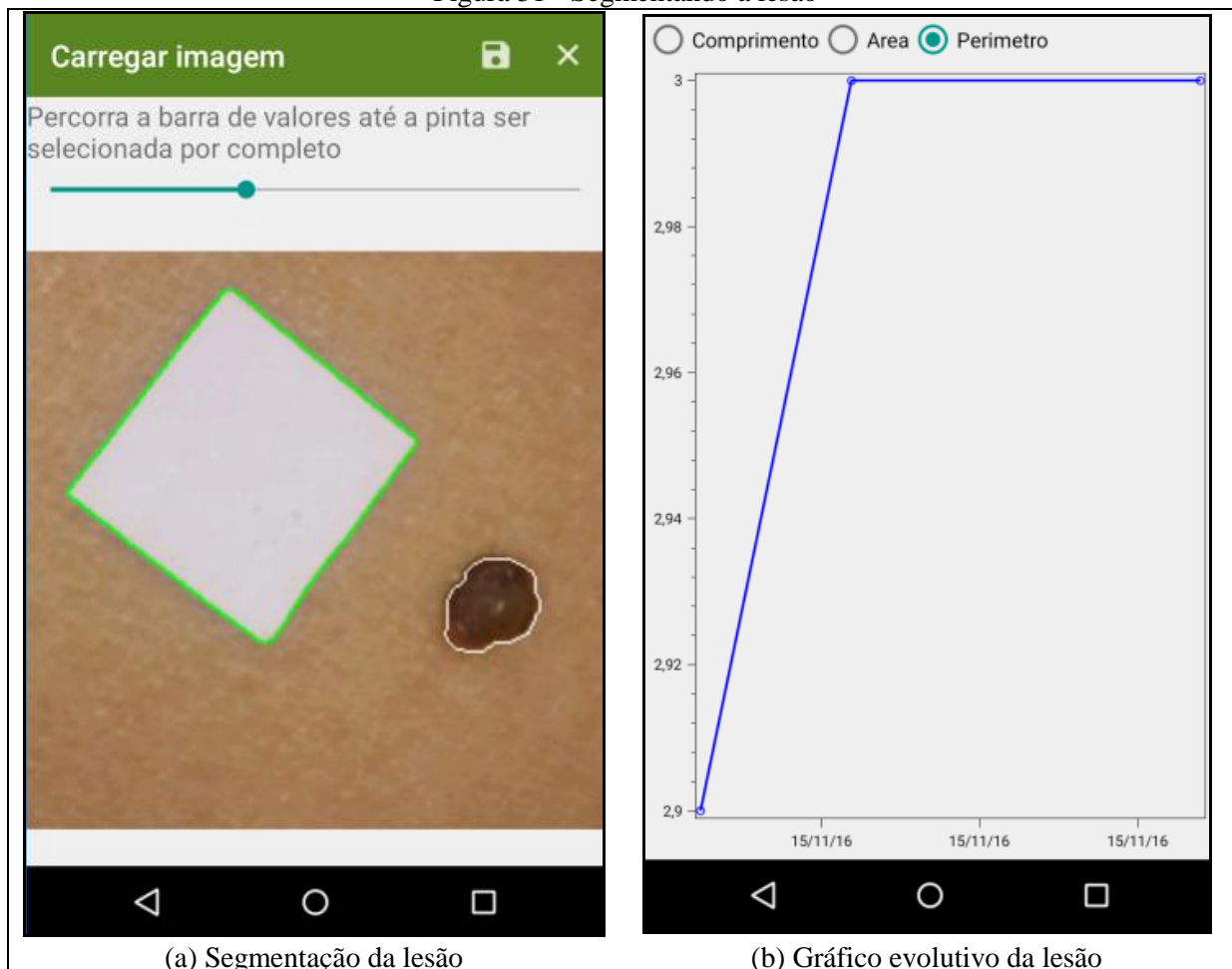


Fonte: elaborado pelo autor.

Ao selecionar uma análise anterior, é realizado o carregamento das informações da análise em questão. Clicando sobre o menu lateral, têm-se as opções *Nova análise* e *Gráfico evolutivo*, conforme apresentado na Figura 30b.

A opção *Nova análise* cria uma nova análise para a lesão. Ao selecioná-la, o aplicativo pede para o usuário de onde ele quer obter a imagem, podendo ser da câmera ou galeria do próprio dispositivo móvel. Selecionada a imagem, o usuário poderá delimitar a região de interesse e, a partir disso, o sistema apresentará o que ele conheceu pinta e gabarito. Caso seja necessário, o usuário poderá ajustar manualmente o valor que melhor isole a lesão. Esta funcionalidade é disponibilizada a partir da tela representada na Figura 31a.

Figura 31 - Segmentando a lesão



Fonte: elaborado pelo autor.

Por fim, o usuário pode salvar a análise através do botão de *salvar* e ao clicar na opção *Gráfico evolutivo* ele poderá ver o histórico de crescimento da lesão na forma de gráfico, conforme mostra a Figura 31b.

### 3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados experimentos realizados com o aplicativo e os resultados obtidos. Na seção 3.4.1 é apresentado à forma que os testes foram aplicados. Por fim, a seção 3.4.2 demonstra os resultados obtidos a partir da proporcionalização das lesões.

#### 3.4.1 Aplicação dos testes

Após ter as imagens carregadas no emulador de dispositivo móvel contendo a aplicação, o teste foi realizado por um usuário. Ele realizou o cadastro das pintas e a análise das mesmas sem precisar de algum tipo de tutorial ou ajuda, apenas sendo informado que a câmera deveria estar reta com a pinta e o gabarito na hora da captura das imagens. Para cadastro dentro do aplicativo o usuário conseguiu realizar o processo totalmente sozinho de forma intuitiva.

Os testes realizados consistem em calcular a área, o perímetro e o comprimento da lesão, com imagens de 10, 20 e 30 centímetros de distância da lesão e do gabarito para verificar se a medição está correta mesmo em distâncias diferentes.

#### 3.4.2 Resultados obtidos nos testes

Na Tabela 1 são apresentados os resultados obtidos a partir do comprimento da lesão. O cabeçalho indica a parte do corpo e ao lado dele o nome da pinta cadastrada no aplicativo juntamente com a medida capturada manualmente.

Tabela 1 - Resultado dos dados obtidos no comprimento da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo / Nuca	Ombro esquerdo / Abaixo da nuca	Ombro esquerdo / Parte esquerda	Peito / Pinta 1	Torso/ Meio
Tamanho real	4mm	6mm	5mm	6mm	5mm	6mm	6mm	5mm
10cm	-	5mm	5mm	6mm	5mm	6mm	6mm	4mm
20cm	4mm	6mm	6mm	6mm	5mm	6mm	5mm	4mm
30cm	5mm	6mm	5mm	6mm	5mm	6mm	6mm	5mm
40cm	ERRO	-	-	-	-	-	-	-

Fonte: elaborado pelo autor.

A partir da Tabela 1 pode-se observar que o protótipo obteve 75% de acertos nas medições. As medições erradas foram de no máximo 1 milímetro de diferença. Ao tentar realizar uma captura numa distância de aproximadamente 40 centímetros o aplicativo não conseguiu identificar o gabarito corretamente, resultando em erro para o usuário.

A Tabela 2 indica o resultado obtido para a o cálculo da área da lesão. Não foi possível obter-se o tamanho da área exato de todas as lesões manualmente. Por isso, os acertos comparados foram baseados na semelhança dos resultados obtidos pela distância da imagem capturada. Apenas a lesão da Coxa direita/Pinta2 foi possível calcular, pois é uma lesão circular e por meio da fórmula da área de um círculo  $\pi \times R^2$  conseguiu-se chegar ao resultado aproximado como indica o cabeçalho.

Tabela 2 - Dados obtidos em relação a área da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo / Nuca	Ombro esquerdo / Abaixo da nuca	Ombro esquerdo / Parte esquerda	Peito / Pinta 1	Torso /Meio
Tamanho real			1,98cm					
10cm	-	1,8cm	2cm	3,1cm	2,2cm	2,1cm	3,2cm	1,4cm
20cm	1,6cm	2,5cm	2,3cm	3,2cm	2,2cm	2,2cm	3cm	1,3cm
30cm	2,1cm	2,4cm	2,3cm	3,1cm	2,2cm	2,6cm	2,9cm	1,4cm
40cm	ERRO	-	-	-	-	-	-	-

Fonte: elaborado pelo autor.

Para a validação da área estabeleceu-se um limite de 2 milímetros com base na outra medição para verificar se ela estava correta. Chegou-se a uma taxa de acerto de 42%. Ao tentar identificar o gabarito na imagem com uma distância de 40 centímetros ocorreu erro e não foi possível realizar a medição.

Os resultados dos cálculos de perímetro obtidos podem ser observados na Tabela 3, onde para verificar se está correto ou não, se utilizou os limites com o valor de 4 centímetros baseado nas outras medições.

Tabela 3 - Dados obtidos em relação ao perímetro da lesão

Distância da captura	Braço direito/ Pinta 1	Coxa direita/ Pinta 1	Coxa direita/ Pinta 2	Ombro esquerdo / Nuca	Ombro esquerdo / Abaixo da nuca	Ombro esquerdo / Parte esquerda	Peito / Pinta 1	Torso/ Meio
10cm	-	4cm	4,2cm	5,5cm	4,5cm	5cm	4,8cm	3,6cm
20cm	3,8cm	4,6cm	4,5cm	5,1cm	4,4cm	5cm	4,5cm	3,4cm
30cm	4,1cm	4,7cm	4,7cm	5,1cm	4,2cm	5,1cm	4,6cm	3,5cm
40cm	ERRO	-	-	-	-	-	-	-

Fonte: elaborado pelo autor.

Ao verificar os valores com a tolerância de 4 milímetros, constatou-se um acerto nas medidas do perímetro de 75%.

As imagens foram capturadas em uma distância de 10, 20 e 30 centímetros, o resultado da captura das imagens juntamente com a segmentação da lesão e gabarito pode ser vistos no Apêndice B.

## 4 CONCLUSÕES

Este trabalho propôs criar um aplicativo para calcular o tamanho e acompanhar a evolução de lesões de peles, tendo uma imagem carregada pelo próprio usuário, a fim de detectar um possível câncer de pele.

Entre os objetivos do trabalho estão a segmentação da lesão e gabarito, proporcionalização da lesão com gabarito resultando no tamanho em medidas reais, e acompanhamento do crescimento das lesões cutâneas.

O desenvolvimento do projeto foi realizado na IDE Visual Studio 2015 utilizando a linguagem C#. Para o desenvolvimento do aplicativo foi utilizado o *framework* Xamarin e a biblioteca EmguCV para Android para o reconhecimento de imagens.

Os resultados do projeto se mostraram satisfatórios. Na detecção do gabarito e lesão o aplicativo obteve um resultado de 95,8%, apenas em uma imagem capturada numa distância de 40 centímetros, não foi possível encontrar o gabarito e a lesão.

Para o objetivo da proporcionalização se obteve um resultado de 75% no cálculo de comprimento da lesão, os outros 25% dos testes tiveram os cálculos errados por diferença de apenas 1 milímetro. O perímetro da lesão também teve um acerto de 75% das lesões, com uma tolerância de 4 milímetros (10% da mediana dos resultados). E o cálculo da área teve um acerto de apenas 42% com tolerância de 2 milímetros (10% da mediana dos resultados).

Conclui-se que caso a segmentação da lesão e do gabarito sejam totalmente corretas (sem a interação do usuário), e também a captura da imagem seja reta em relação à lesão é possível identificar o tamanho real da lesão. Não pode ser usado como diagnóstico médico, pois muitos valores não são exatos, mas para o acompanhamento pessoal é possível utilizar o aplicativo.

### 4.1 EXTENSÕES

Como sugestões para trabalhos futuros propõem-se:

- a) aprimorar a segmentação da lesão;
- b) identificar na representação do corpo o local exato da lesão;
- c) utilizar as outras classificações da regra ABCDE para tentar identificar um possível melanoma;
- d) aprimorar a captura da imagem para focalizar na lesão e no gabarito, assim facilitando o processo de segmentação.

## REFERÊNCIAS

- ABASSI, Naheed R.; SHAW, Helen M.; Rigel Darrell S. Early Diagnosis of Cutaneous Melanoma: Revisiting the ABCD Criteria. **JAMA**. v. 1, n. 292, p. 2771-2776, Dez. 2004. Disponível em: <<http://jama.jamanetwork.com/article.aspx?articleid=199929&resultclick=1>>. Acesso em: 04 abr. 2016
- ABOUD NETA, S. R.; DUTRA, L. V.; ERTHAL, G. J. Limiarização Automática em Histogramas Multimodais. In: CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES, 7. (DINCON)., 2008, Presidente Prudente. **Anais...** 2008. DVD. Disponível em: <[https://www.researchgate.net/profile/Sumaia\\_About\\_Neta/publication/228854150\\_Limiarizacao\\_Automtica\\_em\\_Histogramas\\_Multimodais/links/0912f5133d0c1cc641000000.pdf](https://www.researchgate.net/profile/Sumaia_About_Neta/publication/228854150_Limiarizacao_Automtica_em_Histogramas_Multimodais/links/0912f5133d0c1cc641000000.pdf)>. Acesso em: 02 abr. 2016.
- ALENCAR, Francisco Eduardo S. **Desenvolvimento de um sistema para classificação automática de imagens dermatoscópicas para dispositivos móveis**. 2015. 76 f. Dissertação (Programa de PósGraduação em Ciência da Computação) - Universidade do Estado do Rio Grande do Norte, Mossoró. Disponível em: <<http://ppgcc.ufersa.edu.br/wp-content/uploads/sites/42/2014/09/francisco-eduardo-silva-alencar.pdf>>. Acesso em: 02 abr. 2016.
- ARAUJO, A. F. **Método para extração e caracterização de lesões de pele usando difusão anisotrópica, crescimento de regiões, watersheds e contornos ativos**. 2010. 100 f. Dissertação (Pós-Graduação em Ciência da Computação) - Pós-Graduação em Ciência da Computação, Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto. Disponível em: <[http://repositorio.unesp.br/bitstream/handle/11449/98659/araujo\\_af\\_me\\_sjrp.pdf?sequence=1&isAllowed=y](http://repositorio.unesp.br/bitstream/handle/11449/98659/araujo_af_me_sjrp.pdf?sequence=1&isAllowed=y)> . Acesso em: 02 abr. 2016.
- BOAVENTURA, I.A.G. **Números Fuzzy em Processamento de Imagens Digitais e Suas Aplicações na Detecção de Bordas**. 2010. 218 f. Tese de doutorado - Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18152/tde-06052010-154227/en.php>>. Acesso em: 25 nov. 2016.
- CHAITANYA, Krishna N.; PAVANI, I. Melanoma Early Detection using Dual Classifier. **Internarional Journal of Scientific Engineener and Technology Research**, v. 1, n. 4, p. 1-6, Set. 2015. Disponível em: <<http://ijsetr.com/uploads/315426IJSETR6795-1311.pdf>>. Acesso em: 02 abr. 2016.
- COSTA, Luciano F.; CESAR JR., Roberto M. **Shape analysis and classification: theory and practice**. Boca Raton: CRC Press, 2000.
- CUFÍ, X.; MUÑOZ, X.; FREIXENET, J.; MARTÍ, J. A Review on image segmentation techniques integrating region and boundary information. **Advances in Imaging and Electron Physics**. Eds. HAWKES, P. W., Academic Press, v.120, p.1-39, 2002.
- FACON, J. Morfologia matemática: teoria e exemplos. Curitiba: Champagnat, 1996.
- \_\_\_\_\_. A Morfologia Matemática e suas Aplicações em Processamento de Imagens. In: Workshop de Visao Computacional, 7., 2011, UFPR. **Anais do VII workshop de visao computacional**. Curitiba: Hugo Vieira Neto, 2011. 61-128. Disponível em: <<https://www.ppgia.pucpr.br/~facon/Books/2011WVCMinicurso2Morfo.pdf>> . Acesso em: 25 nov. 2016.

FARIA, Diogo. **Análise e Processamento de Imagem**. 2010. 44 f. Trabalho prático (Mestrado integrado em Engenharia Biomédica) – Mestrado integrado em Engenharia Biomédica, Faculdade de Engenharia da Universidade do Porto, Porto. Disponível em: <[https://web.fe.up.pt/~tavares/downloads/publications/relatorios/MEB\\_Diogo\\_Faria\\_TrabPraticos.pdf](https://web.fe.up.pt/~tavares/downloads/publications/relatorios/MEB_Diogo_Faria_TrabPraticos.pdf)> . Acesso em: 02 abr. 2016.

FEITOSA, Luziana N. **Sistema de contagem automática de ovos do Aedes aegypti a partir de Processamento de Imagens das palhetas de Ovitampas**. 2015. 41 f. Trabalho de conclusão de curso (Graduação em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, Pau dos ferros. Disponível em: <[http://memoria.ifrn.edu.br/bitstream/handle/1044/777/tcc\\_luziana.pdf?sequence=1&isAllowed=y](http://memoria.ifrn.edu.br/bitstream/handle/1044/777/tcc_luziana.pdf?sequence=1&isAllowed=y)> . Acesso em: 04 abr. 2016.

GONZALES, R. C.; WOODS, R. E. **Processamento de imagens digitais**. 2ª. ed. São Paulo: Edgard Blucher, 2000.

INSTITUTO NACIONAL DO CÂNCER. **Pele melanoma**. [S.l.], 2016. Disponível em: <[http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/pele\\_melanoma/definicao+>](http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/pele_melanoma/definicao+>)>. Acesso em: 02 abr. 2016.

JENSEN, Daniel; ELEWSKI, Boni. The ABCDEF Rule. **The Journal of Clinical and Aesthetic Dermatology**, Alabama, v. 2, n. 15, p. 2-8, Fev. 2015. Disponível em: <[http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4345927/#\\_\\_ffn\\_secitle](http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4345927/#__ffn_secitle)>. Acesso em: 02 abr. 2016.

LIMA, Janaína C. M. **Corpus Mapping**: um aplicativo para o mapeamento corporal de lesões cutâneas. 2015. 93 f. Trabalho de conclusão de curso (Graduação em Ciência da Computação) – Centro de Ciências Exatas e Naturais - Universidade Regional de Blumenau, Blumenau. Disponível em: <[http://dsc.inf.furb.br/arquivos/tccs/monografias/2015\\_2\\_janaina-carraromendonca\\_monografia.pdf](http://dsc.inf.furb.br/arquivos/tccs/monografias/2015_2_janaina-carraromendonca_monografia.pdf)>. Acesso em: 02 abr. 2016.

MAIA, D. S. **Deteção e Reconhecimento de Face utilizando o Matlab**. 2014. 70f. Monografia (Graduação em Ciência da Computação) — Universidade Estadual do Sudoeste da Bahia, Bahia. Disponível em: <<http://www2.uesb.br/computacao/wp-content/uploads/2014/09/DETEC%C3%87%C3%83O-E-RECONHECIMENTO-DE-FACE-UTILIZANDO-O-MATLAB-deise-santana-maia.pdf>>. Acesso em: 04 abr. 2016.

MEIRA JR., José; ABBADE, Luciana; STOLF, Hamilton. As múltiplas faces do melanoma cutâneo primário: série de casos. **Diagn Tratamento**, São Paulo, v. 2, n. 20, p. 95-99, Jun. 2015. Disponível em: <<http://files.bvs.br/upload/S/1413-9979/2015/v20n3/a4895.pdf>>. Acesso em: 02 abr. 2016.

MELANOMA RESEARCH FOUNDATION. **The ABCDEs of Melnoma**, 2015. Disponível em: <<http://www.melanoma.org/understand-melanoma/diagnosing-melanoma/detection-screening/abcdes-melanoma>>. Acesso em: 02 abr. 2016.

NEGRI, Rogério G.; SILVA, Marcus F. S. Um novo método de segmentação de imagem com abordagem baseada em bordas e regiões. **Revista Brasileira de Cartografia**, n. XX/YY, p. 1-12, 2012. Disponível em: <[https://www.researchgate.net/profile/Rogério\\_Negri/publication/255820732\\_Um\\_Novo\\_Metodo\\_de\\_Segmentao\\_de\\_Imagem\\_com\\_Abrdagem\\_Baseada\\_em\\_Bordas\\_e\\_Regies/links/00b49520b8f72a2a23000000.pdf](https://www.researchgate.net/profile/Rogério_Negri/publication/255820732_Um_Novo_Metodo_de_Segmentao_de_Imagem_com_Abrdagem_Baseada_em_Bordas_e_Regies/links/00b49520b8f72a2a23000000.pdf)>. Acesso em: 02 abr. 2016.



- NUNES, E. O.; CONCI, A. Texture Segmentation and Edge Location of Regions in Multiband Images. **IEEE Latin America Transactions**, n. 5, p. 184-191, 2007. Disponível em:  
<[https://www.researchgate.net/publication/3455057\\_Texture\\_Segmentation\\_and\\_Edge\\_Location\\_of\\_Regions\\_in\\_Multiband\\_Images](https://www.researchgate.net/publication/3455057_Texture_Segmentation_and_Edge_Location_of_Regions_in_Multiband_Images)>. Acesso em: 25 nov. 2016.
- PAROLIN, Alessandro; HERZER, Eduardo; JUNG, Cláudio. R. **Semi-automated diagnosis of melanoma through the analysis of dermatological images**. In: 23rd SIBGRAPI – Conference on Graphics, Patterns and Images, 23., 2010, Gramado. **Anais...** Rio Grande do Sul: CEGRAPI, 2010. p. 32-38. Disponível em:  
<<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5720349>>. Acesso em: 04 abr. 2016.
- PERUCH, Francesco. **(SEMI)-automated analysis of melanocytic lesions**. 2015. 102 f. Tese de Doutorado (Dipartimento di Ingegneria dell'Informazione) - Universita Degli Studi di Padova, Padova. Disponível em:  
<[http://paduaresearch.cab.unipd.it/7820/1/Thesis\\_Peruch.pdf](http://paduaresearch.cab.unipd.it/7820/1/Thesis_Peruch.pdf)>. Acesso em: 02 abr. 2016.
- PIRES, V. B. **Utilização de Equações Diferenciais Parciais para Eliminação de Ruídos e Detecção de Bordas**. 2008. 140 f. Trabalho prático (Mestrado em ciência da computação) – Universidade Federal de Uberlândia, Uberlândia. Disponível em: <<http://penelope.dr.ufu.br/bitstream/123456789/715/1/UtilizacaoEquacoesDiferenciais.pdf>> . Acesso em: 25 nov. 2016.
- QUEIROZ, J.E.R.; GOMES, H. M. Introdução ao processamento digital de imagens. **Revista RITA**, Campina Grande, v. 8, n. 1, p. 1-31, abr. 2001. Disponível em:  
<<http://www.academia.edu/download/44642121/Rita-Tutorial-PDI.pdf>> . Acesso em: 25 nov. 2016.
- SKIN CANCER FOUNDATION. **Skin cancer facts & statistics**. [S.l.], 2016. Disponível em: <<http://www.skincancer.org/skin-cancer-information/skin-cancer-facts>>. Acesso em: 02 abr. 2016.
- \_\_\_\_\_. **Do you know your ABCDEs**, 2015. Disponível em:  
<<http://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/do-you-know-your-abcde>>. Acesso em: 02 abr. 2016.
- SOARES H. B. **Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte**. 2008. 160f. Tese de Doutorado em Engenharia Elétrica e de Computação, Universidade Federal do Rio Grande do Norte, Natal. Disponível em: <<http://repositorio.ufrn.br:8080/jspui/handle/123456789/15118>>. Acesso em: 02 abr. 2016.
- SURVEILLANCE, EPIDEMIOLOGY AND RESULT PROGRAM (United States). **SEER Stat Fact Sheets: Melanoma of the Skin**. 2012. Disponível em:  
<<http://seer.cancer.gov/statfacts/html/melan.html>>. Acesso em: Acesso em 02 abr. 2016.
- THOMAS, L.; TRANCHAND, P.; BERARD, F.; SECCHI, T. Semiological value of ABCDE criteria in the diagnosis of cutaneous pigmented tumors. **Dermatology**. 1998, v. 197, n. 1, p. 11-17. Disponível em: <<http://www.karger.com/Article/Abstract/17969>>. Acesso em: 04 abr. 2016.

VALE, Giovani M. V;POZ, Aluir P.D. **O processo de detecção de bordas de canny**. In: Anais do Simpósio Brasileiro de Geomática, 2., 2002, Presidente Prudente. **Anais...** São Paulo: UNESP, 2002. P.292-303. Disponível em:<[http://docs.fct.unesp.br/departamentos/cartografia/eventos/2002\\_I\\_SBG/htm/artigos/pdf/A\\_048.pdf](http://docs.fct.unesp.br/departamentos/cartografia/eventos/2002_I_SBG/htm/artigos/pdf/A_048.pdf) >Acesso em: 25 nov. 2016.

## APÊNDICE A – Descrição dos Casos de Uso

Nesta seção são apresentados os detalhamentos dos casos de uso, com descrição, pré-condições, fluxos principais, alternativos, de exceção e pós-condições.

No Quadro 16 apresenta-se o caso de uso UC02 - Carregar imagem.

Quadro 16 - Caso de uso Carregar imagem

UC02	Carregar imagem
Descrição	Este caso de uso tem por objetivo carregar a imagem para analisar
Ator	Usuário
Pré-condição	Estar na tela de nova análise
Fluxo principal	O sistema abre a tela de carregar imagem; O usuário opta por selecionar uma imagem dentro da galeria do dispositivo ou de capturar foto.
Cenário – Capturar foto	UC01 – Capturar imagem
Cenário – Carregar foto	O sistema abre a galeria do dispositivo; Usuário seleciona a imagem; Sistema faz validações sobre o arquivo selecionado; Imagem é carregada.
Cenário – Criação do gabarito	Usuário seleciona a função Criação do gabarito; O sistema abre a tela de como montar o gabarito; Usuário segue instruções.
Pós-condição	Aplicativo carrega a imagem e faz a sua análise

Fonte: elaborado pelo autor.

No quadro 17 apresenta-se o caso de uso UC06 - Segmentar lesão.

Quadro 17 - Caso de uso Segmentar lesão

UC06	Segmentar lesão
Descrição	Este caso de uso tem por objetivo separar a lesão da pele a fim de análise da mesma.
Ator	Sistema
Pré-condição	UC02 – Carregar imagem
Fluxo principal	Sistema realiza verificações da imagem; Sistema realiza a segmentação da imagem; Sistema envia a imagem para análise.
Cenário – Imagem mal informada	Sistema mostra mensagem ao usuário informando que imagem precisa ser recarregada; Sistema abre tela de carregamento de imagem.
Pós-condição	UC07 – Calcular tamanho da lesão

Fonte: elaborado pelo autor.

No Quadro 18 apresenta-se o caso de uso UC07 - Calcular tamanho da lesão.

Quadro 18 - Caso de uso calcular tamanho da lesão

UC07	Calcular tamanho da lesão
Descrição	Este caso de uso tem por objetivo calcular o tamanho da lesão que segmentada
Ator	Sistema
Pré-condição	UC06 – Segmentar lesão UC05 – Segmentar gabarito
Fluxo principal	Sistema realiza cálculo do tamanho da lesão.
Pós-condição	UC08 – Calcular evolução da lesão

Fonte: elaborado pelo autor.

No quadro 19 apresenta-se o caso de uso UC06 - Segmentar gabarito.

Quadro 19 - Caso de uso segmentar gabarito











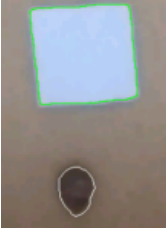

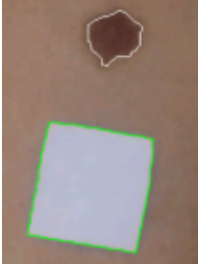










UC05	Segmentar gabarito
Descrição	Este caso de uso tem por objetivo separar o gabarito da pele a fim de análise do mesma.
Ator	Sistema
Pré-condição	UC02 – Carregar imagem
Fluxo principal	Sistema realiza verificações da imagem; Sistema realiza a segmentação da imagem; Sistema envia a imagem para análise.
Cenário – Imagem mal informada	Sistema mostra mensagem ao usuário informando que imagem precisa ser recarregada; Sistema abre tela de carregamento de imagem.
Pós-condição	UC07 – Calcular tamanho da lesão

Fonte: elaborado pelo autor.

**APÊNDICE B – Resultados obtidos**

Nesta seção são apresentadas as imagens obtidas a partir dos testes realizados, demonstrando a identificação do gabarito e da lesão. Essas imagens podem ser visualizadas no Quadro 20.

Quadro 20 - Resultado das análises realizadas

Parte do corpo / Medida	10 centímetros	20 centímetros	30 centímetros
Ombro esquerdo / Abaixo da nuca			
Ombro esquerdo / Parte esquerda			
Peito / Pinta 1			
Torso/Meio			
Braço direito/ Pinta 1			
Coxa direita/ Pinta 1			
Coxadireita/ Pinta 2			
Ombro esquerdo / Nuca			

Fonte: elaborado pelo autor.