



Desenvolvimento de um software para análise de eletrocardiogramas utilizando dispositivos móveis

João Paulo Folador¹

David Calhau Jorge²

Resumo: De acordo com os indicadores de mortalidade da Rede Interagencial de Informações para a Saúde (RIPSA) de 2010, cerca de 30% dos óbitos estão relacionados às doenças do aparelho circulatório. Outro agravante à saúde é a falta de profissionais qualificados na área de doenças cardíacas. Ao verificar estes dados, nota-se uma vasta área para estudos na tentativa de ajudar a diminuir este quadro. Neste contexto, o estudo proposto visa o desenvolvimento de um aplicativo capaz de digitalizar parte de um eletrocardiograma e armazená-lo. Além disso, proporcionar meios para facilitar o compartilhamento rápido e variado destas imagens digitalizadas, realizar o tratamento da imagem capturada de modo a isolar apenas o sinal do eletrocardiograma deixando-o pronto para análise e estudo de possíveis doenças que acometem o coração. O software foi desenvolvido e destinado a dispositivos móveis que utilizem o Android como sistema operacional. Desta forma, há a possibilidade de que os profissionais da área envolvida possam trocar informações e experiências de forma rápida e prática, podendo identificar uma grave doença no paciente.

Palavras-chave: Eletrocardiograma portátil, aplicação Android, processamento de imagens.

¹ UFTM – Universidade Federal do Triângulo Mineiro

² UFTM – Universidade Federal do Triângulo Mineiro

Abstract: According to the Interagency Network of mortality indicators of Health Information (RIPSA) 2010, about 30% of deaths are related to cardiovascular diseases. Another problem to health is the lack of qualified professionals in the area of heart disease. By checking this data, there is a vast area for studies in an attempt to help decrease this fact. In this context, the proposed study aims to develop an application that can scan a portion of an electrocardiogram and store it. In addition, provide means to facilitate rapid and varied sharing of these scanned images, perform the treatment of the captured image in order to isolate only the signal of the electrocardiogram leaving it ready for analysis and study of possible diseases that involve the heart. The software was developed and designed for mobile devices that use the Android operational system. Thus, there is the possibility that the professionals of the area involved to exchange information and experiences quickly and conveniently, and can identify a serious disease in the patient.

Keywords: Portable electrocardiogram, Android application, image processing.

1. Introdução

De acordo com os indicadores de mortalidade da Rede Interagencial de Informações para a Saúde (RIPSA) ilustrados na tabela 1, pode-se perceber os diferentes grupos de causas relacionando os óbitos separados por região do Brasil. Percebe-se que de todas as causas, as doenças do aparelho circulatório são as que mais matam em todas as regiões do país.

Com base neste fato, é perceptível que o grupo de causas das doenças do aparelho circulatório é uma grande área para estudos, visto que, as doenças cardiovasculares são as que mais causam óbitos. No Brasil, a taxa chega a uma média de 30% de mortalidade por este tipo de doença (Soares, 2010).

O eletrocardiograma (ECG) é o tipo de exame mais usado e importante para detectar doenças do coração nos pacientes. (Chen et. al. 2013). Contudo, a classificação dos padrões mostrados nos exames são uma tarefa difícil de ser feita, dado os diferentes tipos de doenças existentes. Os programas de computador podem ajudar a reconhecer estes padrões e disponibilizá-los para que o médico possa avaliá-los. Exames como o *Holter* que captam informações do paciente por longos períodos de tempo (24 horas por exemplo) geram muitas informações que demandam muito tempo de análise. Deste modo, o processamento automático destes exames auxilia e reduz o tempo de resposta ao caso examinado (Benali et. al, 2012).

Tendo em vista os diferentes exames para detecção de várias doenças do coração, a tecnologia vem servir de ferramenta no auxílio à tomada de decisão do médico ou ainda, alertar de um possível problema que o médico não tenha notado num primeiro momento e que possa ser urgente. Foram feitos alguns estudos comparando os resultados publicados de programas de interpretação de ECG, sendo levado em consideração apenas os resultados obtidos e não as metodologias aplicadas. No mesmo estudo, verificou-se alguns pacientes em emergência hospitalar e, se caracterizou a precisão dos diagnósticos entre 57,6 % e 83,0 %, sendo notado, uma variação muito alta nesta precisão. Posteriormente, confirmaram que a eficiência de uso de computadores para auxiliar no diagnóstico era acima de 80% de precisão de acerto. (Salerno et. al. 2003).

Tabela 1: Proporção de óbitos (%) por Região segundo Grupo de Causas.

Grupo de Causas	Região Norte	Região Nordeste	Região Sudeste	Região Sul	Região Centro-Oeste	Total
Doenças do aparelho circulatório	24,84	31,13	31,62	31,06	29,01	30,87
Demais causas definidas	19,04	21,5	20,79	19,19	19,49	20,53
Neoplasias	13,37	14,22	17,66	20,7	15,58	16,93
Causas externas	21,52	16,38	11,04	12,25	17,33	13,55
Doenças do aparelho respiratório	9,78	9,03	12,56	11,53	11,02	11,27
Doenças infecciosas e parasitárias	6,23	4,61	4,64	3,83	5,09	4,62
Afecções originadas no período perinatal	5,22	3,13	1,67	1,44	2,49	2,24
TOTAL	100	100	100	100	100	100

Fonte: Ministério da Saúde/SVS - Sistema de Informações sobre Mortalidade – SIM, 2010.

Além da variação alta na precisão dos diagnósticos, quando se analisa os equipamentos para exames cardíacos e pessoal qualificado, percebe-se que ainda há uma boa quantidade de eletrocardiógrafos analógicos, principalmente na rede pública de hospitais e ainda há uma carência de profissionais qualificados para dar conta da demanda de exames feitos, pondera-se, então, que a tecnologia pode ajudar a encontrar soluções para alguns destes problemas. (Mansur, 2006).

Focado na tentativa de ajudar a diminuir o quadro de óbitos causados pelas doenças cardíacas, amenizar a falta de profissionais e viabilizar o uso de equipamentos antigos, optou-se pelo desenvolvimento de uma aplicação que pudesse atender a estes quesitos. Assim, notou-se que os dispositivos móveis podem auxiliar neste contexto. Portanto, este trabalho busca desenvolver uma aplicação prática e de baixo custo, que seja capaz de digitalizar parte um eletrocardiograma impresso em papel, isolar o sinal do ECG para análise, compactar este sinal isolado utilizando um tipo de compressão e tornar fácil a forma de compartilhar este sinal com profissionais qualificados distantes do centro que necessite de ajuda imediata, e, por fim, disponibilizar um catálogo de doenças cardíacas para consulta rápida dentro do aplicativo. Com isso, é possível facilitar a análise dos exames e combater a elevada taxa de obtidos provocada pelos problemas no sistema circulatório.

2. Metodologia

O trabalho foi delimitado aos últimos 15 anos, em relação às técnicas computacionais envolvendo dispositivos móveis, pois estas são mais recentes. Já os demais assuntos fugindo desde escopo e podendo chegar aos últimos 30 anos de publicação.

O software foi desenvolvido em Java destinado ao sistema operacional para dispositivos móveis Android. O dispositivo usado necessita de cartão de memória para armazenamento dos dados e das imagens, além de precisar de uma câmera para capturar a imagem de parte do eletrocardiograma. A linguagem de programação Java foi escolhida por ser gratuita, possuir grande apoio de comunidades que trabalham com ela e por ser independente de plataforma.

A seguir foram tratadas as características e conceitos fundamentais para alcançar e validar o desenvolvimento de um software com uma interface amigável destinada a capturar informações de imagens de eletrocardiogramas, tratá-las de forma a preservar os sinais necessários e disponibilizá-lo, de forma fácil para transmissão entre dispositivos móveis com a tecnologia Android, além de implementar um método de compactação para transmissão e reconstrução do sinal do ECG.

2.1 Funcionamento básico do coração

Para melhor entendimento sobre o funcionamento do eletrocardiograma, precisa-se conhecer parte da eletrofisiologia do coração. É sabido que 60% de parte de nosso organismo contém água, e nela estão presentes proteínas, glicose, sais minerais como o cloreto de sódio, o cloreto de potássio, o cloreto de cálcio e outros. Estas substâncias são geralmente carregadas de energia eletroquímica, sendo encontrados nos formatos de íons positivos e negativos. Estes íons se movimentam pelas células, e quando este movimento ocorre no coração, estas cargas elétricas produzem uma corrente elétrica de natureza eletroquímica (Azevedo, 1999).

Assim, ao se considerar o estado de repouso, as células do músculo cardíaco possuem suas cargas elétricas positivas e negativas, sendo estas separadas pela membrana celular tendo seu interior negativo. Portanto, a parte externa é positiva e a interna é negativa. Neste caso a célula está em diástole elétrica, veja figura 1, ou polarizada, sendo -90 mV para a parte negativa e +10 mV para a positiva. No momento em que há a inversão destes polos por algum estímulo físico, químico ou biológico, há uma inversão das cargas elétricas, e isto é chamado de sístole elétrica ou despolarização (Azevedo, 1999).

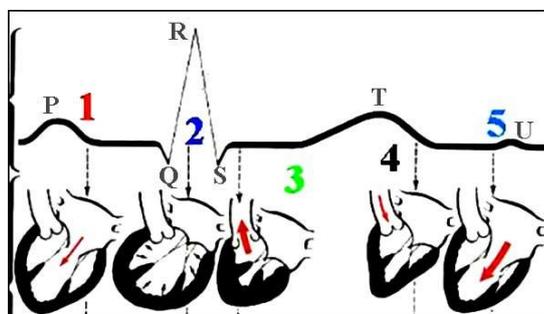


Figura 1: Funcionamento do coração.

Fonte: <http://goo.gl/6Z79V9>

Ainda na figura 1, em (1) há o início da diástole, onde ocorre a abertura das válvulas tricúspide e mitral e o enchimento ventricular; já em (2), ocorre o fechamento das válvulas de entrada,

final da diástole; na marcação (3) é realizada a contração ventricular, abertura das válvulas pulmonar e aórtica - sístole ventricular; (4) é final da sístole ventricular, onde há o fechamento das válvulas pulmonar e aórtica; em (5) ocorre o reinício da diástole atrial e ventricular. Deste modo, a atividade cardíaca pode ser monitorada e caracterizada (Azevedo, 1999).

Com a possibilidade de monitorar a condução elétrica do coração, Einthoven padronizou os registros destes sinais, sendo postulado os três pontos que formavam o triângulo de Einthoven e suas linhas de derivação (D). Na figura 3 (a) tem-se as derivações unipolares das extremidades sendo aVR o potencial obtido no braço direito, aVL obtida do braço esquerdo e aVF da perna esquerda. Essas derivações são precedidas com a letra “a” porque apresentam um potencial elétrico muito baixo e precisam ser ampliadas, por isso da letra “a” de aumentada. A letra “V” significa potencial elétrico e as demais letras vem de **R**ight (direito), **L**eft (esquerdo) e **F**oot (pé) (Azevedo, 1999).

Já na figura 3 (b) está ilustrado as derivações bipolares das extremidades (de Einthoven), caracterizando três derivações: D1, que corresponde à linha que une o braço direito e o braço esquerdo; D2, a linha que vai do braço direito à perna esquerda e D3, que une o braço esquerdo e a perna direita. Estas derivações captam simultaneamente o potencial elétricos entre dois pontos.

Por fim, na figura 3 (c), segundo Azevedo (1999), é possível visualizar as derivações unipolares precordiais (de Wilson), sendo os potenciais elétricos melhor analisados pelo menos em dois planos. Assim, V1, é obtido no quarto espaço intercostal direito junto ao esterno; V2, quarto espaço intercostal esquerdo, junto ao esterno; V3, ponto médio entre V2 e V4; V4, quarto espaço intercostal esquerdo, na linha hemiclavicular; V5, corresponde ao ponto na mesma altura que V4, na linha axilar anterior e V6, o qual reflete o ponto na mesma altura que V4, na linha axilar média. Desde modo, ao se unir estas 6 derivações foi criado as derivações unipolares precordiais, e ao se juntar todas as derivações tem-se doze pontos para análise da atividade elétrica do coração que se juntam e compõem o exame do eletrocardiograma.

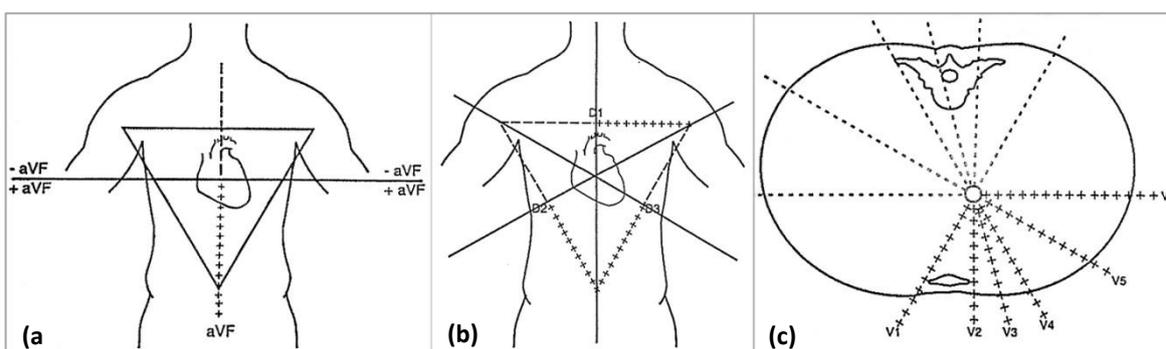


Figura 2: Derivações do coração
Fonte: Adaptado (AZEVEDO, 1999).

2.2 Electrocardiograma

Os primeiros registros dos sinais do coração datam por volta da metade do século XIX. Físicos, fisiologistas e médicos foram autores da descoberta dos eletrodos, item indispensável para verificar a diferença de potencial elétrico no corpo humano e produzir os sinais registrados de um eletrocardiograma até os dias atuais. A partir de testes em pequenos animais, foi possível averiguar a contratatura muscular na presença de corrente elétrica, trabalho este feito por Kollicker e Muller. Posteriormente, em 1887, Waller, por meio do eletrômetro capilar – instrumento desenvolvido por Gabriel Lippmann – conseguiu medir o nível de corrente elétrica do corpo em relação ao coração. (Ginefra, 2007).

Já em 1901, Willem Einthoven conseguiu registrar com eficiência a corrente elétrica gerada pelo coração e a sua propagação pelo corpo humano. Para isso, o pesquisador utilizou pares de eletrodos bipolar para captar a variação da tensão e um galvanômetro de corda para os registros. Os eletrodos nada mais eram do que potes com uma solução salina concentrada, onde o paciente mergulhava as mãos e o pé esquerdo. Esta solução servia como condutor de eletricidade entre a pele e o metal do pote. As duas mãos formavam um par de eletrodo, a mão esquerda e o pé esquerdo formavam o outro par, e o último par era formado pelo braço direito e o pé esquerdo do paciente. (Ginefra, 2007).

Em conseqüente, Einthoven, fez algumas correções matemáticas no sinal registrado por Lippmann e introduziu o conceito dos pontos PQRST no sinal registrado, figura 3. Em 1902, Einthoven consegue registrar o eletrocardiograma de um paciente no hospital universitário a uma distância superior a um quilômetro a partir de seu laboratório usando um cabo telefônico. Este evento, também histórico, Einthoven o chamou de “telecardiograma”.

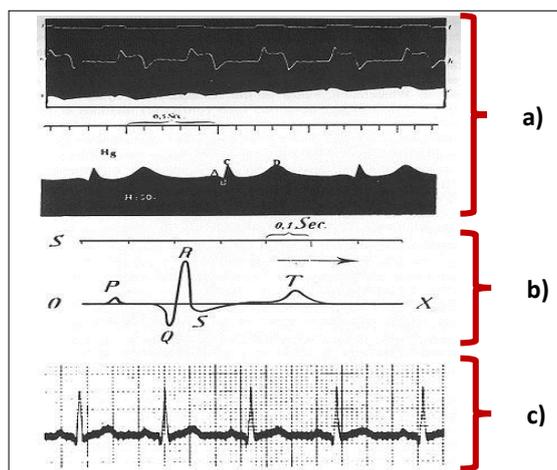


Figura 3: Eletrômetro capilar de Lippmann (a), correção matemática (b), galvanômetro de corda (c).

Fonte: <http://goo.gl/YJu180>

Em 1906, Einthoven publicou um trabalho que mostrava a compreensão sobre o funcionamento do coração e a variedade de anomalias. Após alguns anos, ele desenvolveu a corda de galvanômetro, a qual o permitiu descrever detalhes eletrocardiográficos de doenças acometidas pelo coração. Em 1924, seus anos de dedicação e estudos o consagraram com o prêmio Nobel de

Fisiologia ou Medicina. Desta época em diante, vários trabalhos relacionados com o estudo de eletrocardiogramas tornaram-se foco de estudos.

Atualmente, pode-se dizer que o equipamento responsável em registrar o exame eletrocardiograma é denominado eletrocardiógrafo. Este equipamento passou por várias mudanças desde sua primeira versão onde se tinha uma agulha ou estilete que desenhava um gráfico sobre um rolo de papel em movimento. O papel usado no exame do ECG é milimetrado e termossensível e que, normalmente, se desloca a uma velocidade constante de 25 mm/s. Dentro deste contexto, quando o estilete que registra o sinal do ECG não oscila, há um potencial zero, ou seja, é traçada uma linha reta chamada linha isoeétrica ou linha base. Quando há oscilação desta linha é porque a atividade elétrica está sendo captada e fazendo com que o estilete movimente-se para cima ou para baixo. O papel também é marcado a cada 5 mm tanto na horizontal quanto na vertical, isto é usado para facilitar a visualização das características do exame (Azevedo, 1999).

Os eletrocardiógrafos, atualmente, são encontrados com características digitais, dotados ou não de impressoras de alta resolução, saída para outros dispositivos dentre outras características. Por fim, sabe-se que o eletrocardiograma é um exame no qual o paciente fica em repouso e, geralmente, coloca-se no paciente, em torno de doze eletrodos, nos membros inferiores, superiores, tórax e peito. Assim, é captada a atividade elétrica do coração e enviado ao equipamento que repassa este sinal, no formato de um gráfico impresso em um papel, o qual, posteriormente, serve para análise das sequências dos batimentos. Podendo assim, caracterizar um tipo de doença. (Cury, et al. 2011).

2.3 Digitalização do eletrocardiograma

O exame do eletrocardiograma é feito e impresso, em sua maioria, em tiras de papel representando os sinais colhidos dos batimentos cardíacos do paciente. Para digitalizar este sinal, de forma prática, pode-se utilizar um scanner ou uma câmera digital. Neste trabalho, por ser focado na mobilidade, praticidade e rapidez, optou-se pelo processo de fotografar o exame.

Quando se captura uma imagem por meio de uma câmera ela pode apresentar algumas distorções de posicionamento, de iluminação, baixa resolução da câmera usada, má qualidade do papel que contém o sinal do eletrocardiograma dentre outros. Portanto, é preciso tratar parte destes problemas.

As câmeras dos dispositivos móveis em geral possuem alguns parâmetros que podem ser configurados para um melhor aproveitamento e diminuição de alguns problemas como iluminação, resolução, tipo de fotografia dentre outros.

Os parâmetros de configuração da câmera podem ser escolhidos a partir de uma classe chamada *Camera.Parameters*, onde é possível ajustar vários parâmetros da câmera. Após estes ajustes, o usuário deve fazer a chamada do método *setParameters* para confirmar estas características. A configuração escolhida, para este projeto, foi optar pela câmera em preto e branco,

eliminando assim os tipos diferentes de coloração da grade de fundo dos eletrocardiogramas. Do mesmo modo, os parâmetros de foco da imagem e iluminação por meio do flash foram configurados em modo automático. Já a resolução da câmera foi pré-definida na resolução intermediária do dispositivo, ou seja, é feita uma busca no hardware do dispositivo e listado quais os tipos de configurações disponíveis, posteriormente é escolhida a resolução intermediária. (Google, 2014),

Quando se digitaliza a imagem do eletrocardiograma por meio de uma câmera, os fatores descritos anteriormente são então utilizados na tentativa de amenizar e melhorar a imagem que posteriormente será processada ajustando-a para os formatos necessários na análise da imagem e transmissão para outros dispositivos. Nos próximos tópicos serão discutidos os diferentes tipos de filtros utilizados para o processamento digital do eletrocardiograma.

2.3.1 Imagem digital

Quando se fotografa uma determinada cena é necessário armazenar as informações que a representa de alguma forma. No caso das máquinas fotográficas antigas haviam alguns processos químicos e mecânicos para armazenar a cena em um plástico fotossensível, o negativo, que posteriormente era revelado. Já nas máquinas fotográficas digitais, muito comuns e fáceis de encontrar em praticamente todos os aparelhos celulares, a cena capturada pela câmera é armazenada na memória do dispositivo ou em cartões de memória. Segundo Queiroz e Gomes (2001), a imagem digital é composta por um conjunto de bits dispostos de forma bidimensional, e cada elemento desta matriz pode ser chamado pixel, veja figura 4.

A leitura desta matriz é feita através do posicionamento da linha (m), depois pelo índice da coluna (n). A representação na tela desta imagem é convencionalmente no quarto quadrante, e os índices aumentam da esquerda para direita e do topo da tela para a parte inferior. Assim sendo, a representação do par ordenado $f(x, y)$ denota a informação contida em um pixel na matriz de representação da imagem digital.

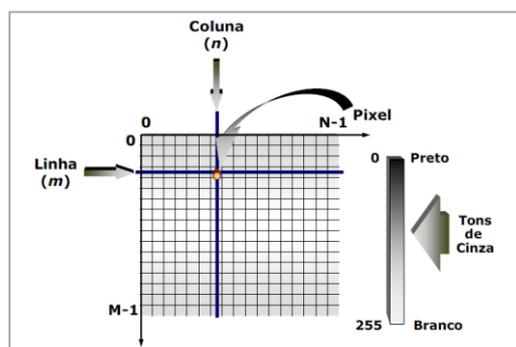


Figura 4: Arranjo da imagem digital.
Fonte: Queiroz, Gomez (2001).

Contudo, a representação destes bits saíram de um sinal analógico, onde cada valor de x tem uma representação $f(x)$. Deste modo, como o computador entende apenas 0s e 1s, é preciso encontrar os valores discretos deste sinal contínuo (o sinal capturado pelo sensor de imagem). Para isso, uma técnica de amostragem é usada para representar o sinal contínuo, e posteriormente é feita a quantização que é o processo de atribuir valores para a amplitude do sinal amostrado, veja o exemplo ilustrado na figura 5. (Scuri, 2002).

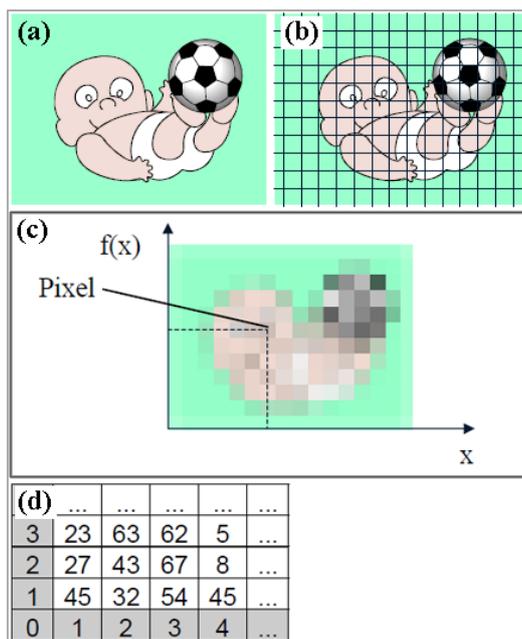


Figura 5: Processo de digitalização da imagem, em (a) tem-se a imagem contínua, (b) a amostragem, (c) a quantização e em (d) a matriz codificada da imagem.
Fonte: SCURI (2002).

Nota-se que, ainda na figura 7 letra (c), a representação da imagem depois de digitalizada ficou muito ruim, quase impossível de identifica-la, mas se aumentar a quantidade da amostragem a resolução da imagem final aumentará proporcionando uma melhor representação da imagem original. Assim sendo, os próximos tópicos tratarão as relações diretas dos algoritmos que manipulam estas matrizes de pixels a ponto de obter as melhores características de filtragem desejadas.

2.3.2 Interpolação bilinear

A interpolação bilinear é uma das técnicas básicas encontradas para alterar o tamanho de uma imagem. Assim sendo, quando uma imagem precisa sofrer uma transformação de tamanho, esta

técnica calcula os níveis de pixels de saída em escala de cinza por meio de uma distância ponderada dos níveis de cinza dos quatro pixels ao redor do ponto desejado da imagem de entrada. Este processo é uma extensão da interpolação linear. Deste modo, a figura 6 ilustra a interpolação que trabalha em duas direções e tenta encontrar o melhor valor de cor e intensidade dos pixels ao redor do pixel escolhido. (Efford, 2000).

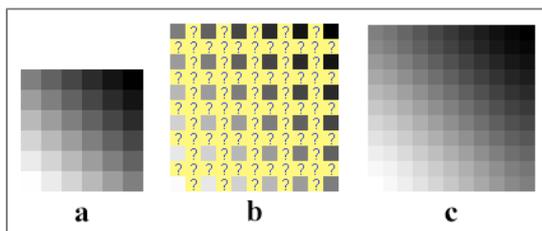


Figura 6: Exemplo de interpolação bilinear.

Fonte: Adaptada (MCHUGH, 2014).

A equação 1 representa o cálculo necessário para encontrar o novo valor em escala de cinza de um pixel de saída.

$$\begin{aligned}
 f(x, y) = & f(0, 0) * (1 - x) * (1 - y) \\
 & + f(1, 0) * x * (1 - y) + f(0, 1) * (1 \\
 & - x) * y + f(1, 1) * x * y
 \end{aligned}
 \tag{1}$$

Segundo Efford (2000), sabe-se que quanto menor a distância do pixel a ser interpolado em relação ao seu vizinho, mais influenciado por este ele será e, conseqüentemente, influenciará na sua cor final. Existem outros métodos de redimensionamento de imagens como: *nearest neighbor*, o qual utiliza um cálculo simples baseado no vizinho mais próximo, considerando assim apenas um pixel vizinho. Este método causa um “serrilhamento” na imagem redimensionada, por este motivo não se optou pelo seu uso. Também existem técnicas mais sofisticadas, como: interpolação bicúbica, que utiliza uma vizinhança de 4x4, aumentando assim a qualidade da nova imagem. Contudo, exige um poder computacional maior para sua execução (Petrou, 2010).

2.3.3 Imagem invertida

Normalmente uma imagem comum é dita imagem positiva, e quando se deseja invertê-la o valor de cada pixel é submetido a uma subtração do valor máximo da profundidade da imagem, ou seja, para uma profundidade de 8 bits tem-se uma variação de 0 a 255, sendo então utilizado o valor máximo 255. Este tipo de filtro é comumente conhecido como negativo. A equação (2) denota o valor em escala de cinza do novo pixel P', o qual teve seu valor subtraído de 255. Aplicar o negativo a uma dada imagem ajuda a discriminar alvos ajudando no processo de identificação de características, como por exemplos, em imagens médicas. (Queiroz, Gomes, 2001).

$$P'(x, y) = 255 - P(x, y) \quad (2)$$



Figura 7: Aplicação do filtro negativo.

Fonte: (QUEIROZ, HERMAN, 2001).

Na figura 7, tem-se a ilustração de uma imagem original em (a) e a aplicação do filtro negativo da imagem em (b).

2.3.4 Threshold

O filtro *Threshold* basicamente faz a transformação de uma dada imagem para binário usando o valor limite especificado. Todos os pixels com intensidades iguais ou superiores ao valor limiar são convertidos em pixels brancos. Todos os outros pixels com intensidades abaixo do valor limiar são convertidos em pixels pretos. Este tipo de efeito é normalmente usado quando se deseja uma abordagem envolvendo segmentação de imagens. Matematicamente seria representado como segue na equação (3).

$$F'(x, y) = \begin{cases} 1 & \text{se } F(x, y) > T \\ 0 & \text{se } F(x, y) < T \end{cases} \quad (3)$$

Portanto, para este tipo simples de *threshold* se o valor do limiar (T) fosse 140, por exemplo, então todos os pixels acima deste valor receberiam 1, enquanto os valores abaixo de 140 ficariam com 0. Existem várias técnicas que utilizam a abordagem de segmentação de imagens. A seguir serão explicados mais algumas formas de aplicar o *threshold*.

2.3.5 Blob filtering

Esta técnica é embasada em um método matemático de detecção de regiões em imagens digitais que busca encontrar na imagem os pixels que possuem características semelhantes, como brilho ou cor, e comparando-as com as regiões ao redor. Um *blob* é uma região da imagem que apresenta característica constante ou varia dentro de um range definido, ou seja, dentro de um mesmo *blob* considera-se que seus pixels são todos similares. Neste contexto, as tarefas envolvendo visão

de máquinas o *blob detection* é uma técnica muito usada, como: rastreamento de uma bola vermelha, detectar a pele de uma pessoa e movimentos da mão.

Neste sentido é possível caracterizar e conseguir informações sobre as regiões da imagem, sendo que por meio de outras técnicas como detecção de borda ou de cantos não seria possível. Assim, depois de detectar a região pode-se fazer a busca por reconhecimento de objetos. Um dos tipos de *blob detection* mais comuns é o *Laplacian of Gaussian (LoG)*, onde se utilizam uma matriz como *kernel* para aplicar uma equação matemática, a qual resulta em respostas positivas para *blob* escuros e respostas negativas para *blobs* claros. Outros tipos de filtros são: *the determinant of the Hessian*, este utiliza uma matriz quadrada denominada matriz de Hessian para suavizar a seleção do *blob*; *affine-adapted differential blob detectors*, refere-se a uma técnica mais elaborada que trabalha com formas invariantes de translação, rotação e escala no domínio espacial para detecção e seleção dos *blobs* (Gonzalez; Woods, 2007).

Segundo Wang, Ren e Liu (2008), alguns passos são executados para buscar as manchas na imagem. O valor do range do pixel, neste processo, deve estar entre 0 e 255 caracterizando o tom de cinza da imagem. Após a transformação da imagem em escala de cinza, é feito um processamento da matriz de dados no sentido de encontrar os *blobs*. Depois é feito alguns cálculos para levantar informações do *blob*, como o tamanho e o valor da coordenada do ponto central. Em consequente, a função de detecção de *blobs* busca identificar os pixels com o mesmo nível de cinza, e estes pixels são então diferenciados e separados baseados no critério de interconexão entre eles. Por fim, a coordenada do centro do *blob* pode ser calculada de acordo com a equação (4).

$$x_{center} = \int_A x dA / A, y_{center} = \int_A y dA / A \quad (4)$$

Assim, o centroide do *blob* é calculado e, esta informação será utilizada para, por exemplo, delinear a área do *blob*.

2.3.6 Extract biggest blob

Este filtro utiliza parte das técnicas já mostradas no tópico anterior, contudo a ideia fundamental deste filtro é buscar na imagem uma mancha (*blob*) grande e extrair ela, ou utilizar a localização deste *biggest blob* para realizar operações diversas. Normalmente, utiliza-se uma imagem de fonte para encontrar as manchas e realizar as operações pertinentes em uma outra imagem cópia para não danificar a imagem original. (Queiroz, Gomes, 2001).

Os passos utilizados em um algoritmo para extrair os *blobs* poderiam ser:

- Buscar as regiões da imagem que, de alguma forma, são coerentes;
- Marcar os grupos destas regiões;
- Extrair a mancha maior da imagem.

A figura 8, ilustra a aplicação do filtro para extrair o maior *blob*.

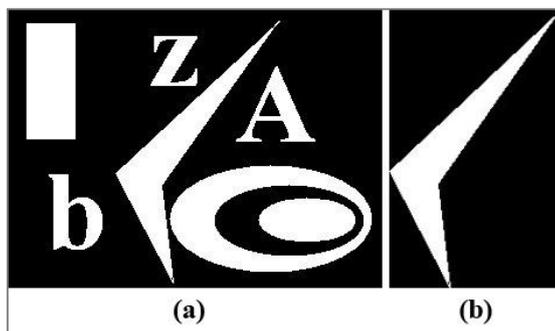


Figura 8: Em (a) tem-se a imagem original, e (b) o resultado após extrair o maior objeto.

Fonte: AForge.NET (2012).

2.4 O cenário dos dispositivos móveis

Segundo Lecheta (2010), cerca de 3 bilhões de pessoas possuem aparelho celular, e cada vez mais estes indivíduos procuram celulares que apresentem melhores características de hardware e facilidade de uso, ou seja, aqueles que possuem câmeras, meio de comunicação via Wi-fi e bluetooth, boa interface, que executem bons jogos, dentre outros. E com essa facilidade e mobilidade, as empresas buscam aproveitar ao máximo as funcionalidades de aplicativos desenvolvidos para os dispositivos móveis que tragam maior agilidade na tomada de decisão e/ou na inserção de informações nos servidores das empresas.

Por outro lado, empresas buscam lucrar no desenvolvimentos destes tipos de aparelhos, focados nas características almejadas pelos usuários, estes fabricantes de dispositivos móveis tem que criar equipamentos cada vez mais fáceis de usar e com poder computacional maior. E, neste contexto, existem inúmeros fabricantes líderes no mercado, como: Samsung, Motorola, LG, Sony Ericsson, Nokia, Nexus, dentre outras empresas.

A empresa Google se juntou com vários destes fabricantes para criar um padrão de desenvolvimento e plataforma de código aberto para os dispositivos móveis. Este grupo ficou conhecido como *Open Handset Alliance* (OHA).

Segundo a AG2 Publicis Modern (2013), a qual fez um trabalho objetivando mostrar algumas características e os hábitos de utilização de smartphones, constatou-se que as funções mais usadas são: ligações, mensagens enviadas via app, SMS, redes sociais e o despertador. Depois vem a utilização de jogos. Ainda foi notado que 61% dos usuários utilizam o aparelho logo após acordarem, 78% dos aparelhos não possuem aplicativos pagos, 43% mexem no celular enquanto assistem TV e que 70% dos usuários deitam-se em suas camas e continuam mexendo no celular. Outra ponto importante é que cerca de 26% dos brasileiros possuem smarthphones, já nos Estados Unidos este número chega a 56% da população. E em relação aos diferentes tipos de plataformas que gerenciam os aparelhos, 25% dos smartphones possuem as plataformas: symbian OS, Windows

Phone e outras. Já o IOS da empresa Apple chega a 24% do mercado mundial. Por fim, o Android fica com 51% da fatia do mercado de smartphones no mundo.

Por fim, pode-se perceber a variedade de características usadas dos smartphones e que a plataforma Android domina mais da metade do mercado mundial. Contando com estes dados do mercado, plataforma de código aberto e gratuito, a base da linguagem de desenvolvimento Java e grandes comunidades de discussões de projetos, optou-se por utilizar o Android para criar a aplicação foco deste trabalho.

3. Resultados

Após analisar os passos necessários desde a captura da imagem até a limpeza do sinal do eletrocardiograma, optou-se pela verificação de algo já implementado. Contudo, em um primeiro momento encontrou-se muitos trabalhos aplicados utilizando a ferramenta matlab, outros com implementação para desktop e pouquíssimo para dispositivos móveis voltados para o sistema circulatório. Posteriormente, encontrou-se um conjunto de bibliotecas denominadas Catalano Framework, desenvolvida por Marcos Diego Catalano e disponibilizado de forma open source e voltada para programação Java e Android. Este framework tem várias funções que podem ser utilizadas nos projetos de processamento de imagens.

Sendo assim, fez-se uso desta biblioteca como auxílio na construção do projeto, e como hardware para testes utilizou-se basicamente um celular S3 mini da Samsung e um outro celular também do mesmo fabricante denominado Grand Duos, ambos com processador acima de 1 GHz, memória RAM de 1 GB e armazenamento de 8 GB em cartão micro SD. O Grand Duos possui tela de 5 polegadas e câmera de 8 megapixels, ao passo que o S3 mini possui tela de 4 polegadas e câmera de 5 megapixels.

a. Tela principal da aplicação

Para montar a tela principal do aplicativo optou-se por um layout linear dentro de uma estrutura chamada scrollView, que é um tipo de interface visual que pode ter uma barra de rolagem lateral à medida que o conteúdo se expande (Lecheta, 2010).

A figura 9 ilustra cada ícone que representa um botão de entrada em cada parte do software, as quais serão explicadas nos tópicos subsequentes. Portanto, no botão digitalizar ECG é onde o usuário deve entrar para capturar parte da imagem de um eletrocardiograma. Em filtrar, é possível limpar a imagem anteriormente digitalizada, de modo a retirar a grade de fundo e preservar o apenas o sinal em formato binário para futuras análises. Em restaurar ECG o usuário pode remontar o sinal a partir de um arquivo texto transmitido com os pontos comprimidos pela transformada de Fourier. Com o botão compartilhar, o profissional responsável pelo eletro pode enviar a parte do ECG

digitalizado para outro profissional por meio de mensagens de texto com o ECG em anexo, ou e-mail, whatsapp dentre outras formas disponíveis no sistema Android.



Figura 9: Tela principal e diagrama de classe.

Fonte: Autor, 2015.

Ainda na figura 9 está representado o diagrama de classes para construção desta tela. A classe MainActivity contém os atributos do tipo imageView, os quais recebem a montagem do layout de cada ícone presente nesta tela, assim como os eventos que representam as ações do clique do usuário na opção desejada. Já a classe SingleMediaScanner serve para criar a chamada nativa dos recursos do sistema de compartilhamentos do Android. (Google, 2014).

b. Captura do ECG

O processo de captura do eletrocardiograma se dá por meio da câmera fotográfica do dispositivo móvel. Sendo assim, foi necessário o desenvolvimento de uma tela específica para esta tarefa. Para se criar uma tela ou *activity* no Android, basicamente é necessário no mínimo dois arquivos: um arquivo xml que contém o layout da tela e uma classe java que controla e manipula os eventos.

Num primeiro momento foi decidido codificar o uso padrão do hardware da câmera no Android, onde existe uma classe para esta finalidade: `android.hardware.Camera`. Com esta classe é possível realizar a chamada de uma tela já existente no próprio Android que disponibiliza a câmera pronta para fotografar. Exige um pouco de codificação e ajustes do layout da tela para uso simples no projeto. (Lecheta, 2010).

Contudo, posteriormente viu-se a necessidade de implementar algumas funcionalidades para facilitar e guiar o usuário quando utilizasse este recurso, e conseqüentemente, ajudaria no desempenho da aplicação em outros processos. Para que estas características fossem implementadas

foi necessário desenvolver uma mini aplicação para utilizar a câmera do dispositivo móvel, veja figura 10, onde fosse permitido desenhar uma guia na tela do usuário, escrever alguma informação pertinente, salvar a imagem em um local específico e realizar o corte da imagem fotografada pelo usuário dentro destas guias de apoio.

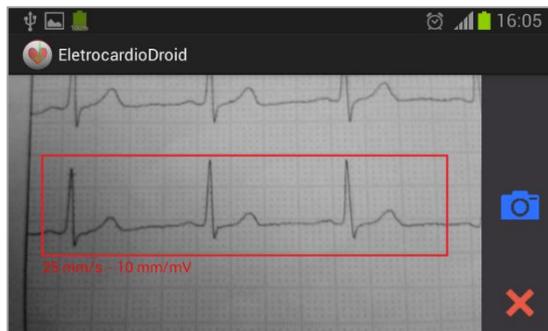


Figura 10: Digitalização do eletrocardiograma.

Fonte: Autor, 2015.

Na figura 10, o usuário deve então posicionar, dentro do retângulo vermelho, a parte do eletrocardiograma que ele deseja compartilhar e/ou analisar, o ícone que representa a câmera fotográfica em azul serve então para tirar a foto e o ícone vermelho no formato de um “x” serve para que ele feche a tela da câmera. Quando a imagem for capturada, ela é cortada com a mesma proporção do retângulo e salva.

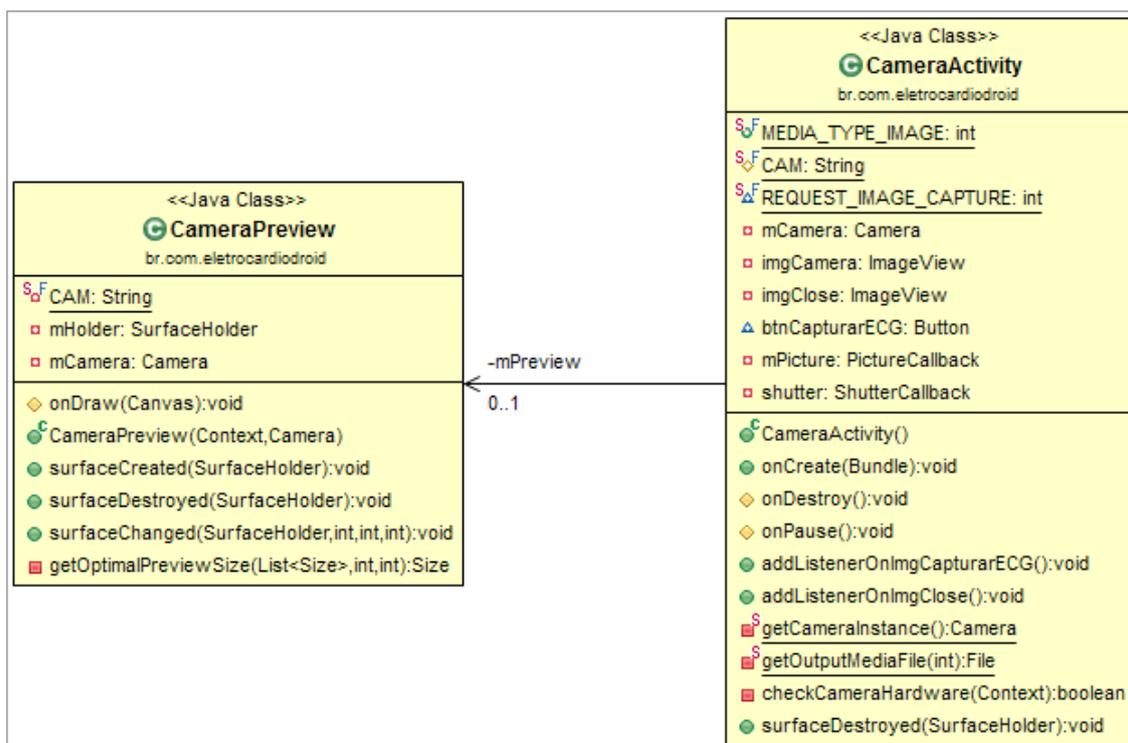


Figura 11: Diagrama de classe da tela de captura do ECG.

Fonte: Autor, 2015.

Na figura 11, de acordo com Google Inc. (2014) a classe CameraPreview estende a classe surfaceView nativa do Android, a qual fornece uma superfície de desenho que pode ser alterada,

redimensionada, ou seja, ajustada de acordo com as necessidades da aplicação. A `CameraPreview` ainda implementa a interface `SurfaceHolder.Callback` que serve para retornar informações de alteração que ocorrem na superfície da câmera. Os métodos que se destacam na classe `CameraPreview` são:

- `onDraw` – permite realizar desenhos e escrever textos por meio do parâmetro `canvas`;
- `CameraPreview` – método construtor da classe, que inicializa a variável `mCamera` e `mHolder`, que recebem uma instância da câmera e do `surfaceHolder` que controla as alterações da superfície, respectivamente;
- `surfaceCreated` – este método é imediatamente chamado após a criação da superfície, configurando onde será desenhado a superfície da câmera e também inicia a visualização da câmera;
- `surfaceDestroyed` – chamado quando a superfície é destruída, antes ele para a visualização da câmera e a libera para outras aplicações;
- `surfaceChanged` – método chamado assim que acontece qualquer alteração de tamanho ou formato na superfície. Nesta etapa também é feito o ajuste da taxa de aspecto da tela (*aspect ratio*) para conseguir um melhor ajuste na tela quando acontece a mudança na superfície, e isso é feito pelo outro método denominado `getOptimalPreviewSize`.

Já a classe `CameraActivity` faz a extensão da classe `Activity`, esta fornece as características pertinentes para criar uma interface, uma tela para interação com o usuário (Google, 2014). Esta tela utiliza o layout criado no arquivo xml, que contém um *framelayout* para posicionar o local onde ficará a câmera e um *gridlayout* para os botões laterais, veja figura 12, e implementa também os métodos relativos aos eventos dos botões e os outros necessários para salvar a imagem capturada. A seguir são comentados os métodos mais importantes:

- `onCreate`: este é método que dá início na configuração da tela. É onde há chamada dos métodos dos eventos dos botões e aqueles que necessitam de configuração, como a câmera, a qual foi configurado ao auto foco, a flash automático, a cor em escala de cinza e a resolução da câmera a ser utilizada. Também é neste método que é feita a chamada de um método auxiliar, `getCameraInstance`, que verifica se existe câmera no dispositivo e faz a requisição ao hardware para sua abertura e atribuindo assim a instância na variável `mCamera`;
- `onDestroy`: libera o hardware da câmera ao finalizar a tela;
- `onPause`: caso haja a interrupção de outra aplicação no dispositivo móvel, este método para a visualização da câmera;
- `addListenerOnImgCapturarECG`: este método está associado ao clique do botão de captura da imagem. Ao capturar a imagem é feito corte da mesma na marcação feita e mostrada

anteriormente na figura 13. Este método também utiliza um outro método auxiliar: *getOutputMediaFile*, este último criar o diretório “eletrocardiogramas” e também a saída do arquivo da imagem com extensão .jpg. O nome do arquivo começa com ECG_ mais um valor gerado aleatoriamente e, por fim, concatenado a extensão da imagem para evitar que as imagens sobrescrevessem uma a outra. Ex.: ECG_165487901.jpg;

- `addListenerOnImgClose`: este método executa o evento do botão fechar da câmera;



Figura 12: telas para filtrar as imagens. Em (a) deve se escolher o ECG, (b) ajustes dos algoritmos de filtragem e aplicação, e em (c) o resultado comparando com a imagem original.

Fonte: Autor, 2015.

Deste modo, esta mini aplicação serve para fotografar o eletrocardiograma e, conseqüentemente, o digitaliza para ser processado por outras partes que compõem o projeto.

c. Filtrar a imagem

Para ajustar a imagem digitalizada do ECG verificou-se a importância em retirar a grade de fundo do eletro deixando apenas o sinal, isso reduz a quantidade de ruídos, melhora o tamanho da imagem para transferência e facilita outros tipos de processamento, como aplicação de transformada de Fourier, além da análise do sinal do ECG por uma rede neural artificial. Assim sendo, após a digitalização o usuário segue para esta parte de filtragem, onde ele escolhe o eletrocardiograma que deseja aplicar os filtros, ajusta os filtros e visualiza na própria tela o resultado da filtragem, podendo rapidamente reajustar os valores para melhorar e aplicar novamente na imagem, veja figura 12 (Garg et. al, 2012). O usuário deve escolher um dentre os eletrocardiogramas digitalizados na parte superior da tela (a) e em seguida tocar na imagem ampliada para seguir para o ajuste dos filtros em (b). Neste passo, existem alguns filtros importantes para extrair o fundo da imagem. Para facilitar a remoção da grade de fundo do ECG é importante converter a imagem para escala de cinza ou binário (Shen, Laio, 2009). Neste sentido, como já foi citado, a câmera é configurada para escala de cinza, mas caso o

dispositivo móvel não tenha este efeito na câmera, ao aplicar os filtros a imagem é convertida para escala de cinza.

O limiar (*threshold*) e a filtragem de gotas (*Blob Filtering*) são configurados, veja figura 12 (b), tocando a suas respectivas barras (*seekbar*) e fazendo o movimento de arrastar. Os valores destas barras foram pré-configuradas em 142 para *threshold* e 63 para *Blob Filtering*, pois analisando algumas fotografias feitas e testando estes filtros, fez-se uma média dos valores para facilitar sua aplicação. Já nas caixas de marcação (*checkbox*), tem-se os filtros: *Extract Biggest Blob*, que imprescindível para extrair o sinal do ECG como um todo, logo o destacando do fundo da imagem ao final do processo; o Refinar que é responsável por deixar apenas um pixel por coordenada (x, y). A figura 13 ilustra o diagrama de classe responsável por esta parte da aplicação.

Na figura 13, a classe *ImageAdapter* estende a classe nativa do Android chamada *BaseAdapter* que permite criar uma tela com visualização de itens preparados em uma matriz. O método *getItem(int)* retorna a posição do item selecionado, já o *getView()* e seus parâmetros montam o objeto de visualização de cada item da lista com altura e largura definida, ou seja, para cada imagem de um dado diretório, este método cria uma pequena imagem de visualização em miniatura para que o usuário possa escolher, veja figura 12 (a). A classe *Galeria.java* é a implementação da tela em si, ou seja, em seu método *onCreate* o atributo *imageList* é preenchido com as imagens lidas do diretório que contém os eletrocardiogramas digitalizados, depois prepara o evento para pegar a imagem selecionada, que posteriormente é passada como parâmetro para ser filtrada. (Google, 2014).

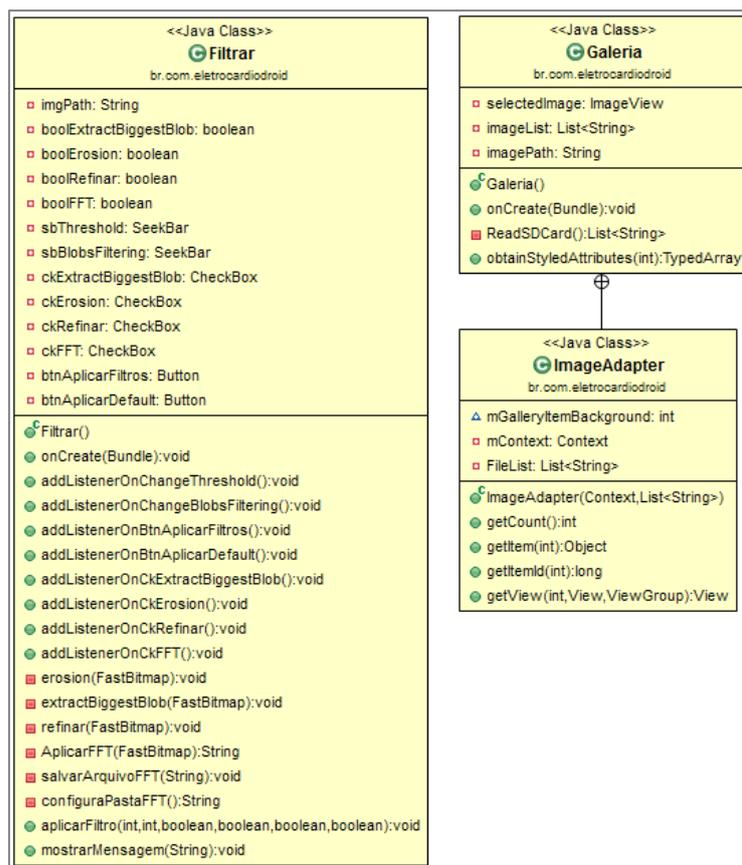


Figura 13: Diagrama de classe do processo de filtragem do ECG.

Fonte: Autor, 2015.

A classe `filtrar.java` é onde acontece a aplicação de todos os algoritmos de filtragem da imagem. De forma simples, os atributos desta classe terão a seguinte finalidade:

- A variável `imgPath`, deve conter o caminho da imagem que antes foi passado via parâmetro da classe `galeria`, onde o usuário fez a escolha do ECG;
- Todas as variáveis que começam com `bool` (`boolExtractBiggestBlob`, `boolErosion`, `boolRefinar`, etc) são do tipo booleano para controlar se este tipo de algoritmo será ou não utilizado.

Os demais atributos são para controle dos eventos de cada componente da tela, por exemplo, `sbThreshold` recebe uma instância do objeto `seekbar` para controlar as alterações deste elemento, `ckRefinar`, serve para configurar o layout de um `checkbox` na tela e controlar seus eventos e assim por diante. Todos os métodos que começam seus nomes por “`addListenerOn...`” referem à criação do objeto relacionado a uma parte específica do layout e adição do evento que espera a ação do usuário. (Google, 2014).

O método `aplicarFiltro(int, int, bool, bool)` é o mais complexo, sendo passados dois parâmetros do tipo inteiro para ele, o primeiro está relacionado ao valor do *threshold* e o segundo ao *blobFiltering*, anteriormente comentados. Depois tem-se dois parâmetros booleanos para

marcar a utilização dos filtros escolhidos, sendo: `extractBiggestBlob` e `refinar`, respectivamente.

Deste modo, os passos que compõem as ações deste métodos são:

1. A imagem escolhida é aberta e decodificada para um objeto no formato bitmap, por meio do método “`BitmapFactory.decodeFile`”;
2. Posteriormente, é feita uma cópia desta imagem sendo redimensionada pela metade de seu tamanho original, diminuindo a carga de processamento, logo a imagem original é liberada da memória;
3. Esta nova imagem é transformada em escala de cinza;
4. Neste passo, a imagem é invertida aplicando o efeito negativo, ou seja, a cor de cada pixel é subtraída de 255;
5. Após a inversão, é aplicado o Limiar sendo padrão igual 142 ou então ajustado pelo usuário. O `threshold` é aplicado valores maiores que o limiar passam a ser 1 e os abaixo passam a ser 0. Nesta etapa a imagem é binarizada;
6. O `blob filtering` segue logo após o `threshold` para eliminar os grupos de pixels isolados que sobraram do grid no fundo da imagem;
7. Depois é verificado se foi marcada a opção para extrair a maior mancha (`extractBiggestBlob`), que retirar o maior objeto da imagem, ou seja, o sinal do ECG;
8. Mas adiante, o `refinar` é verificado, caso esteja selecionado um algoritmo encontrará o ponto médio de cada coluna de pixels do sinal, portanto restará apenas um pixels por coordenada (x, y);
9. Por fim, a imagem resultante é salva no celular em uma pasta denominada `eletrocardiogramas`, com o nome do arquivo da imagem começando por “`resultado_`” e concatenado com um data e hora da geração da mesma. Esta figura tem o formato `.png`. A imagem resultante é então mostrada na tela do dispositivo para visualização do usuário.

4. Discussão dos resultados

A figura 14 ilustra a digitalização de uma parte de um eletrocardiograma que foi impresso em papel e submetido ao programa desenvolvido neste projeto.

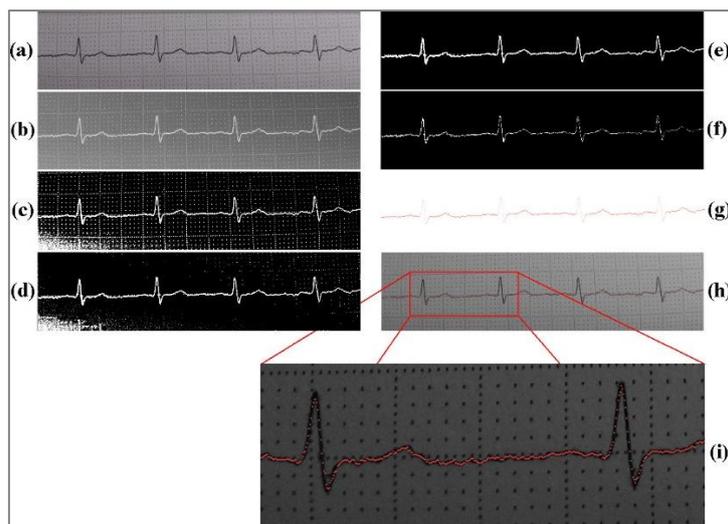


Figura 14: Composição passo-a-passo da execução dos filtros usáveis pela aplicação EletrocardioDroid.
Fonte: Autor, 2015.

Ainda analisando a figura 25, tem-se em (a) a imagem original digitalizada a partir de um ECG impresso em papel com fundo milimetrado. Já em (b), há a aplicação do filtro *invert* ou imagem invertida que também é comumente chamado de negativo, o qual foi comentado anteriormente. Em (c) a imagem passou pelo filtro *Threshold*, que faz o binário da imagem a partir de um limiar. A etapa (d) da figura 24 corresponde à aplicação do filtro *Blob Filtering*, sendo este para remover os conjuntos de pixels isolados menores. Em (e) há a aplicação do *Extract Biggest Blob* que retorna o maior conjunto de pixels que no caso já é o sinal do ECG. Na sub imagem (f) é a ilustração da aplicação do filtro erosão, o qual desbasta o sinal deixando-o mais fino, e este é um filtro que pode ou não ser usado pelo usuário. Em (g) tem-se o resultado final da aplicação dos filtros para remoção do fundo da imagem e isolamento total do sinal do eletrocardiograma. Nesta etapa é aplicado ao sinal o refinar, onde é feita a média dos pixels na vertical de cada coluna da matriz de pixels, deixando assim, apenas um pixel por coordenada x e y. Em (h) é montado o sinal extraído do ECG sobre a imagem original, facilitando a visualização do resultado conseguido. Por fim, na figura 25 (i), o resultado ampliado da etapa (h) é mostrado para melhor compreensão dos pontos sobre o sinal.

Assim, o resultado da extração do sinal do eletrocardiograma feito neste trabalho por meio do “EletrocardioDroid” foi bastante satisfatória, pois os pontos ficaram todos dentro do sinal original, como mostra a figura 25 (i), e quando acontece de algum ponto ficar fora, imediatamente o usuário do programa pode refazer o processo apenas reajustando os parâmetros até que fique o mais adequado possível. O programa EletrocardioDroid já realiza os cálculos automáticos destes parâmetros para execução dos filtros, contudo ainda é possível reajustá-los manualmente. Assim, a ideia proposta por Garg et. al. 2012, o qual utiliza filtragem de mediana para retirar os ruídos das imagem digitalizada e segmentação por cores para retirar a grade de fundo do ECG, sendo os algoritmos criados baseados na linguagem Pascal e destinados a computadores comuns. Quando

se compara aos resultados de Garg et. al. 2012 com a ideia proposta neste trabalho com o EletrocardioDroid, pode-se concluir que foi bastante eficaz a extração da grade de fundo do ECG e ajustes de filtros realizados num dispositivo móvel com a plataforma Android, pois este consegue realizar todas as operações com eficiência, rapidez e praticidade.

5. Conclusão

Os resultados demonstram que a proposta inicial de criar um aplicativo capaz de digitalizar parte de um exame de ECG impresso em papel e extrair a grade fundo é viável em um dispositivo móvel que utilize a plataforma Android. Além disso, o aplicativo desenvolvido obteve um bom desempenho na execução dos algoritmos, sem apresentar lentidão exagerada e/ou travamentos.

Assim, após esta parte do trabalho objetiva-se, em trabalhos futuros, implementar o compartilhamento destes eletrocardiogramas por meios distintos, como: bluetooth, e-mail, what's up dentre outros formatos ágeis. Outro ponto importante a ser criado é uma forma de reconstrução do sinal enviado em formato de pontos em arquivo texto, diminuindo consideravelmente o tamanho do arquivo enviado.

6. Referências

- AZEVEDO, D. F. 1999. *Iniciação à Eletrocardiografia*. Porto Alegre: editora Artes médicas Sul.
- BENALI, R.; REGUIG, F. B.; SLIMANE, Z. H. 2012. *Automatic classification of heartbeats using wavelet neural network*. *Journal of Medical Systems*, v. 36, n. 2, p. 883-892. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/20703646>>. Acesso em: 10 de Jan. 2014
- BONATTI, I. S.; LOPES, A.; PERES, P. L. D.; AGULHARI, C. M. 2013. *Linearidade em Sinais e Sistemas*. Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas. Disponível em: <<http://www.dt.fee.unicamp.br/~peres/LSS.pdf>>. Acesso em: 10 de nov. de 2014.
- CHEN, Y., DING, J., HUANG, C., HO, Y., HUNG, C. 2013. *ECG baseline extraction by gradient varying weighting functions*. *Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Asia-Pacific, vol., pp.1-4. Disponível em: <URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6694129&isnumber=6694103>>. Acesso em: 20 maio de 2014.
- CURY, L. K., SIQUEIRA, E. B., GOMES, T. S. 2011. *Um software para a análise de eletrocardiograma (ECG)*. *Revista CEPPG – Centro de Ensino Superior de Catalão*, Ano XIV, n. 24, p. 187-198.
- EFFORD, N. 2000. *Digital Image Processing: a Practical Introduction Using Java*. Estados Unidos: Pearson Education.
- GARG, D. K., Thakur, D., Sharma, S., Bhardwaj, S. 2012. *ECG Paper Records Digitization through Image Processing Techniques*. *International Journal of Computer Applications*. v. 48, n. 13. Disponível em: <<http://research.ijcaonline.org/volume48/number13/pxc3880485.pdf>>. Acesso em: 20 de Jan. 2014.
- GINEFRA, P. 2007. *A Evolução do Eletrodo no Registro dos Potenciais Elétricos Cardíacos: Um pouco de história*. Disponível em: <<http://www.rbconline.org.br/artigo/a-evolucao-do-eletrodo-no-registro-dos-potenciais-eletricos-cardiacos-um-pouco-de-historia/>>. Acesso em: 15 de out. de 2014.
- GONZALEZ, R. C., WOODS, R. E. 2007. *Digital Image Processing*. United States: Prentice Hall.
- GOOGLE Inc. 2014. *Android Guide: How to build apps using Android's various APIs*. Disponível em: <<http://developer.android.com/guide/index.html>>. Acesso: 01 de Mar. de 2014.
- LECHETA, R. 2010. *Google Android: Aprenda a criar aplicativos para dispositivos móveis com o Android SDK*. São Paulo: Novatec.
- MANSUR, P. H., LACORDAIRE, K. P. C., DESTRO-FILHO, J. B., RESENDE, E. S., DESTRO, J. P., OLIVEIRA, L. M., MORAES, D. C. G.; FREITAS, G. R., ROCHA, L. S. 2006. *Análise de registros eletrocardiográficos associados ao infarto agudo do miocárdio*. *Arq. Bras. Cardiol.* vol. 87, n. 2, pp. 106-114.
- QUEIROZ, J. E., GOMES, H. M. 2001. *Introdução ao Processamento Digital de Imagens*. Departamento de Sistemas e Computação, UFCG. *Revista RITA*. v. 7, n. 1.
- RIPSA, 2013. *Rede Interagencial de Informações para a Saúde*. Disponível em: <<http://tabnet.datasus.gov.br>>. Acesso em: 14 de maio de 2013.
- SCURI, A. E. 2002. *Fundamentos da imagem digital*. Pontifícia Universidade Católica do Rio de Janeiro. Disponível em: <<http://www.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>. Acesso em: 11 de out. de 2014.
- SHEN, T.W., LAIO, T.F. 2009. *Image processing on ECG chart for ECG signal recovery*. *Computers in Cardiology*, pp.725-728, 13-16. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5445282&isnumber=5445250>>. Acesso em: 05 de fev. de 2014.
- SOARES, F. 2013. AdNews. São Paulo. *Pesquisa revela dados sobre o uso de smartphones*. Disponível em: <<http://adnews.com.br/tecnologia/pesquisa-revela-dados-sobre-o-uso-de-smartphones>>. Acesso em: 30 de fev. 2014.