



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

VINICIUS NUNES MACIEL

**VISUALIZADOR DICOM PARA O AUXÍLIO EM DIAGNÓSTICO MÉDICO
POR IMAGEM**

**Assis/SP
2016**

VINICIUS NUNES MACIEL

**VISUALIZADOR DICOM PARA O AUXÍLIO EM DIAGNÓSTICO MÉDICO
POR IMAGEM**

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação.

Orientando: Vinicius Nunes Maciel

Orientador: Prof. MSc. Guilherme de Cleve Farto

**Assis/SP
2016**

FICHA CATALOGRÁFICA

MACIEL, Vinicius Nunes

Visualizador DICOM para o auxílio em diagnóstico por imagem / Vinicius Nunes Maciel Fundação Educacional do Município de Assis – FEMA – Assis, 2016. 85p.

Orientador: Prof. MSc. Guilherme de Cleve Farto
Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. DICOM 2. Visualização de imagens 3. C++ 4. VTK

CDD: 001.6
Biblioteca da FEMA

VISUALIZADOR DICOM PARA O AUXÍLIO EM DIAGNÓSTICO MÉDICO POR IMAGEM

VINICIUS NUNES MACIEL

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. MSc. Guilherme de Cleva Farto

Examinador: Prof. Dr. Luiz Carlos Begosso

Assis/SP

2016

DEDICATÓRIA

Dedico este trabalho à Deus, a toda minha família e minha esposa, em especial meus pais, por terem me dado todo carinho e melhor educação possível e serem um grande exemplo de pessoas. Dedico também aos meus amigos e pessoas que estiveram sempre me apoiando para a realização deste meu sonho.

AGRADECIMENTOS

Primeiramente agradeço a **Deus**, autor e princípio da vida, pois sem Ele, não estaríamos aqui vivendo e desfrutando de momentos tão marcantes e preciosos.

Agradeço à minha mãe **Zilda Nunes Maciel** e meu pai **Ismael Moreira Maciel** que estão sempre ao meu lado me apoiando, orientando e me formando para a vida.

A minha esposa, **Amanda Nogueira Maciel**, por me animar sempre, com seu otimismo deixando meus dias mais alegres e belos.

A todos meus **amigos, parentes e familiares**, que me ajudaram de forma direta ou indireta, pois sem a ajuda deles, sem suas orações e apoio não conseguiria sozinho concluir esta etapa de minha vida.

Ao meu amigo e Professor Me. **Felipe Alexandre Cardoso Pazinato**, por me orientar no início do trabalho e colaborar nesta fase acadêmica tão importante.

Ao meu amigo e Professor MSc. **Guilherme de Cleva Farto**, por me orientar no restante do trabalho e colaborar nesta fase acadêmica tão importante.

A todos os meus **professores** destes quatro anos de curso, que ajudaram em minha formação não só acadêmica, mas também como pessoa.

A todos meus **amigos de classe**, pelo apoio, amizade e companheirismo.

E por fim agradeço a todos que colaboraram de forma direta ou indiretamente para a execução deste trabalho.

“O que você sabe não tem valor; O valor está no que você faz com o que sabe.”

Bruce Lee (1940 - 1973)

RESUMO

Imagens médicas seguem um padrão internacional para que sejam utilizadas em um único formato digital. Esse padrão denomina-se Comunicação de Imagens Digitais na Medicina (DICOM). Por meio de um arquivo DICOM é possível trabalhar com imagens médicas em diferentes equipamentos de imagem e computadores. Este trabalho apresenta a proposta de desenvolvimento de um visualizador de imagens DICOM para o auxílio em diagnósticos feitos por estudos de imagens médicas. O visualizador é desktop, desenvolvido da IDE Visual Studio 2013 na linguagem de programação C++, utilizando a *framework* Qt e as bibliotecas ITK e VTK, disponibilizando funcionalidades mínimas para o uso em situações práticas, tais como: visualização de imagens, seleção de imagens, anotações de texto, exibição de *tags*, aplicação de *zoom*, exportação de imagem e impressão.

Palavras-chave: Processamento de imagens; DICOM; C++; ITK; VTK; QT.

ABSTRACT

Medical Images follow an international standard to be used in a single digital format. This standard is called Digital Imaging and Communications in Medicine (DICOM). Through a DICOM file you can work with medical images in different imaging equipment and computers. This paper presents a proposal to develop a viewer of DICOM images to aid in diagnosis made by medical imaging studies. The viewer is desktop, developed the Visual Studio 2013 IDE in the C ++ programming language using the Qt framework and ITK and VTK libraries, providing minimum functionality for use in practical situations, such as preview images, image selection, notes text, display tags, zooming, export imaging and printing.

Keywords: Image processing; DICOM; C++; ITK; VTK; QT.

LISTA DE ILUSTRAÇÕES

Figura 1: Etapas de um Sistema de Processamento de Imagens (In: SILVA, 2013)	18
Figura 2: Representação de uma imagem digital (In: SANCHES, 2009)	19
Figura 3: Ilustração do histograma de níveis de cinza de uma imagem (In: MONTEIRO, 2005).....	20
Figura 4: Imagem com bom Contraste e seu Respectivo Histograma (In: MONTEIRO, 2005).....	20
Figura 5: Imagem com baixo contraste e seu Histograma (In: MONTEIRO, 2005)	22
Figura 6: Representação do corpo de um arquivo DICOM (In: RIBEIRO, 2013)	35
Figura 7: Representação dos elementos de dados (In: NEMA, 2016)	37
Figura 8: Modelo bi-normal de distribuição de probabilidade para análise por curva ROC (In: AZEVEDO-MARQUES, 2001)	44
Figura 9: Tipos de Planos de Aquisição de Imagens (In: MANSSOUR; FREITAS, 2002)	48
Figura 10: Diagrama de Casos de Uso.....	51
Figura 11: Diagrama de Atividades Abrir Imagem	57
Figura 12: Diagrama de Atividades Exibir Tags.....	58
Figura 13: Diagrama de Atividades Aplicar zoom	58
Figura 14: Diagrama de Atividades Aplicar Contraste	59
Figura 15: Diagrama de Atividades Aplicar Rotação	59
Figura 16: Diagrama de Atividades Aplicar Medida	60
Figura 17: Diagrama de Atividades Exportar Imagem	60
Figura 18: Diagrama de Atividades Imprimir Imagem	61
Figura 19: EAP, representação dos processos envolvidos no projeto	62
Figura 20: Construindo arquivos com CMake	64
Figura 21: Construindo arquivos com Visual Studio	65
Figura 22: Interface da Tela Principal	71
Figura 23: Visualização de imagem DICOM e suas informações	72
Figura 24: Visualização de série de imagens DICOM e suas informações.....	72
Figura 25: Exibição das tags do arquivo	73
Figura 26: Visualização na orientação Sagital	74
Figura 27: Visualização na orientação Coronal	74
Figura 28: Visualização com contraste aplicado.....	75
Figura 29: Visualização com zoom aplicado.....	75
Figura 30: Tela de Laudos	76
Figura 31: Visualização com rotação aplicada.....	76
Figura 32: Visualização com medida aplicada.....	77

LISTA DE TABELAS

Tabela 1: Lista de tags DICOM (In: RIBEIRO, 2013)	37
Tabela 2: Cabeçalho usado pelo padrão DICOM (In: NEMA, 2016)	40
Tabela 3: Representação de algumas das possíveis VRs do Padrão.....	41
Tabela 4: Representação Doença x Resultado (In: PORTO, 2010).....	43
Tabela 5: Abrir Imagem.....	52
Tabela 6: Visualizar Imagem.....	52
Tabela 7: Selecionar fatias.....	52
Tabela 8: Exibir tags.....	53
Tabela 9: Aplicar zoom.....	53
Tabela 10: Aplicar Contraste.....	54
Tabela 11: Aplicar Rotação.....	54
Tabela 12: Aplicar Medida.....	55
Tabela 13: Exportar Imagem.....	55
Tabela 14: Gerar Laudo.....	56
Tabela 15: Imprimir Imagem	56

SUMÁRIO

1 – INTRODUÇÃO	15
1.2 OBJETIVOS	16
1.3 JUSTIFICATIVA.....	16
1.4 MOTIVAÇÃO	16
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO	17
1.6 ESTRUTURA DO TRABALHO	17
2 – REVISÃO BIBLIOGRÁFICA	18
2.1 PROCESSAMENTO DE IMAGENS (DPI).....	18
2.1.1 Representação de Imagens Digitais.....	18
2.1.2 Histograma	19
2.1.3 Realce	20
2.1.4 Contraste	21
2.1.5 Equalização	22
2.1.7 Limiarização (Threshold).....	22
2.2 O PADRÃO DICOM	23
2.2.1 Histórico	23
2.2.2 Partes do Padrão	24
2.2.3 Conformidade (Aderência ao Padrão).....	28
2.2.3.1 Declaração de Conformidade.....	28
2.2.3.1.1 <i>Visão geral da seção de redes para declaração de conformidade.....</i>	<i>29</i>
2.2.3.1.2 <i>Visão geral da seção de armazenamento de mídias para declaração de conformidade.</i>	<i>29</i>
2.2.3.2 Requisitos de conformidade.....	30
2.2.3.2.1 <i>Requisitos de conformidade em redes de computadores</i>	<i>31</i>
2.2.3.2.2 <i>Requisitos de conformidade em intercâmbios de mídias DICOM</i>	<i>31</i>
2.2.4 Estrutura.....	32
2.2.4.1 Definições utilizadas pelo DICOM	32
2.2.4.2 O arquivo DICOM.....	35

2.2.4.2.1 Elemento de Dados.....	35
2.2.4.2.2 Cabeçalho de um arquivo DICOM.....	37
2.2.4.2.3 Representação de Valor (VR)	41
2.3 DIAGNÓSTICO AUXILIADO POR COMPUTADOR (CAD)	41
2.3.1 A Curva ROC	42
2.3.1.1 Sensibilidade e Especificidade	43
2.3.2 Erro de diagnóstico radiológico e percepção	45
2.3.3 Exercício ilegal da radiologia e diagnóstico por imagem	45
2.4 AS IMAGENS MÉDICAS.....	46
2.4.1 Obtenção das imagens médicas	46
2.4.1.1 Os não invasivos.....	47
2.4.1.2 Os invasivos.....	48
2.5 TECNOLOGIAS UTILIZADAS.....	48
3 – METODOLOGIA.....	50
3.1 LISTA DE EVENTOS	50
3.2 DIAGRAMA DE CASOS DE USO	51
3.3 ESPECIFICAÇÃO DE CASOS DE USO	51
3.3.1 Abrir Imagem.....	51
3.3.2 Visualizar Imagem.....	52
3.3.3 Selecionar fatias	52
3.3.4 Exibir Tags	53
3.3.5 Aplicar zoom	53
3.3.6 Aplicar Contraste	54
3.3.7 Aplicar Rotação.....	54
3.3.8 Aplicar Medida	55
3.3.9 Exportar imagem.....	55
3.3.10 Gerar laudo.....	56
3.3.11 Imprimir Imagem	56
3.4 DIAGRAMA DE ATIVIDADES	57
3.4.1 Abrir Imagem.....	57

3.4.2 Exibir <i>Tags</i>	58
3.4.3 Aplicar <i>zoom</i>	58
3.4.4 Aplicar Contraste	59
3.4.5 Aplicar Rotação.....	59
3.4.6 Aplicar Medida	60
3.4.7 Exportar imagem.....	60
3.4.8 Imprimir Imagem	61
3.5 ESTRUTURA ANALÍTICA DO PROJETO (EAP)	61
4 – DESENVOLVIMENTO	63
4.1 CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO.....	63
4.2 IMPLEMENTAÇÃO	66
4.2.1 Método de visualizar imagem e séries de imagens DICOM	66
4.2.2 Método de leitura de imagem e séries de imagens DICOM e extração de <i>tags</i>	68
4.3 INTERFACES DA APLICAÇÃO	71
5 – CONCLUSÃO.....	78
5.1 – TRABALHOS FUTUROS.....	78
REFERÊNCIAS	79
APÊNDICE A	81

1 – INTRODUÇÃO

A tecnologia vem auxiliando o ser humano em distintas áreas, sendo amplamente utilizada na área médica, em que diversos procedimentos e estudos saíram do processo manual para a utilização de equipamentos mais sofisticados. No caso de alguns exames, diagnosticados de maneira demorada e por meio de observações com o paciente, hoje, com o uso da tomografia computadorizada, em apenas alguns minutos o médico pode elaborar ou direcionar o diagnóstico. Tais equipamentos seguem um padrão de formatação das imagens de diagnóstico médico, agrupando a comunicação e o armazenamento das informações médicas em um único formato eletrônico que é estruturado a partir de um protocolo. Com base nestes protocolos, as informações associadas às imagens geradas por equipamentos de tomografias, mamografias, radiografias, ressonâncias magnéticas e outros, obedecem várias regras. O formato eletrônico é denominado *Digital Imaging and Communications in Medicine* (DICOM).

Atualmente, existem softwares que trabalham com a plataforma DICOM como, por exemplo, o OsiriX, porém este possui limitações referentes a não ser um software nacional, possuir um custo impeditivo e não fornecer suporte no mercado brasileiro, além de estar disponível apenas à plataforma Apple. Embora a medicina tenha evoluído ao longo destes anos, ainda há muitas coisas que precisam ser feitas para melhorá-la, como no caso de exames de diagnóstico por imagem. Diagnósticos feitos sem auxílio de software têm chances de erros maiores. Hoje, exames por imagens ainda são conduzidos por impressões em chapas radiológicas. As chapas são caras para o hospital e contêm elementos tóxicos que podem contaminar o meio ambiente, se descartados de maneira incorreta.

A proposta deste trabalho é a criação de um visualizador de imagens no padrão DICOM, possibilitando que as imagens sejam visualizadas em duas dimensões 2D, bem como permitindo ao médico gerar o laudo e imprimir as imagens em impressoras comuns, a partir de um computador, otimizando os diagnósticos feitos pelo médico.

1.2 OBJETIVOS

Este trabalho tem por objetivo desenvolver um visualizador DICOM, para visualização de qualquer tipo de exame por imagem, auxiliando médicos e radiologistas nos diagnósticos de pacientes, com as seguintes funções:

- Utilizar as imagens DICOM para gerar visualizações;
- Visualizar em 2D e de diferentes pontos de vista;
- Aplicar filtros corretivos;
- Gerar laudo do paciente;
- Imprimir imagens em impressoras.

1.3 JUSTIFICATIVA

Diagnosticar pacientes por meio de *Computer-aided diagnosis* (CAD ou Diagnóstico auxiliado por computador) reduz as chances de erro no diagnóstico. Assim, evita-se que o indivíduo doente seja diagnosticado como não doente. Desta forma, a utilização de um visualizador de imagens de exames, no apoio ao diagnóstico, proporciona uma melhor visualização e análise das imagens, evitando que tais situações aconteçam.

1.4 MOTIVAÇÃO

Poucos hospitais adotam os recursos de CAD para diagnosticar pacientes. Assim, a taxa de erros em diagnósticos ainda é considerada alta pela falta de softwares para o auxílio nos hospitais. Esse trabalho poderá contribuir nessa área tão delicada e importante que é a medicina, auxiliando médicos a salvar muitas vidas.

1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

Após a conclusão desta pesquisa, pretende-se publicá-la na forma de artigos e divulgá-la em instituições de ensino com o objetivo de compartilhar os conhecimentos adquiridos ao longo dos trabalhos conduzidos. Se possível, compartilhar com a própria instituição no curso de medicina, para o uso dos alunos e professores. O visualizador possibilitará que diagnósticos de exames por imagem, sejam feitos com o auxílio do computador, com o objetivo de contribuir na área médica e com futuros projetos.

1.6 ESTRUTURA DO TRABALHO

Este trabalho será estruturado nas seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo, contextualiza a área de estudo e apresenta os objetivos, a justificativa e a motivação para o desenvolvimento desta pesquisa.
- **Capítulo 2 – Revisão Bibliográfica:** Neste capítulo, aborda-se a fundamentação teórica necessária para a compreensão desta pesquisa.
- **Capítulo 3 – Metodologia:** Neste capítulo, apresentam-se os requisitos, a especificação, as ferramentas e técnicas utilizadas na implementação e também a operacionalidade do visualizador.
- **Capítulo 4 – Desenvolvimento:** Neste capítulo, aborda-se a configuração e implementação do sistema e imagens das interfaces.
- **Capítulo 5 – Conclusão:** Neste capítulo, apresentam-se os prós e contras, vantagens e desvantagens do visualizador DICOM para o auxílio em diagnósticos de exames.

2 – REVISÃO BIBLIOGRÁFICA

2.1 PROCESSAMENTO DE IMAGENS (DPI)

Em paralelo com as aplicações espaciais, técnicas de processamento de imagem digital começaram a ser investigadas no final dos anos 1960 e início dos anos 1970 para serem adotadas em imagiologia médica, controle remoto, observações de recursos da terra e astronomia (GONZALEZ; WOODS, 2008).

O objetivo de um sistema de processamento de imagens é retirar de um arquivo toda a informação existente, para tal resultado possa ser analisado por uma pessoa comum. Neste sentido, tal atividade desenvolveu técnicas para se chegar ao resultado esperado, como a própria análise da imagem e técnicas de melhoria da mesma, para seu estudo ser facilitado. A Figura 1 ilustra um diagrama que representa as etapas de um sistema de processamento de imagens.

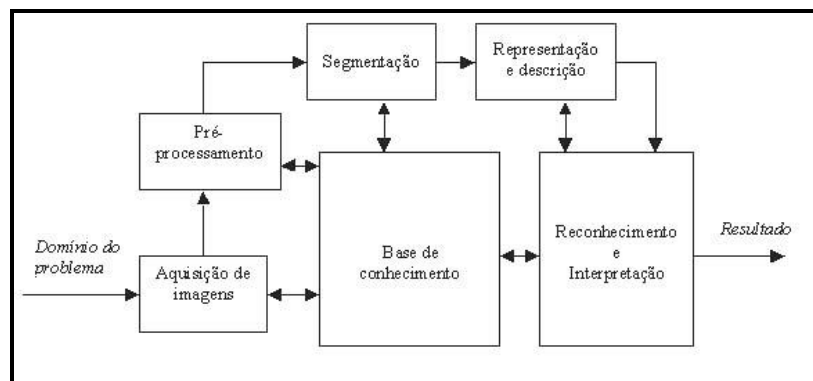


Figura 1: Etapas de um Sistema de Processamento de Imagens (In: SILVA, 2013)

2.1.1 Representação de Imagens Digitais

Uma imagem digital linearizada ou monocromática é uma função bidimensional $f(x, y)$ que representam intensidades luminosas, onde x e y são coordenadas espaciais, que por convenção $x = [1, 2, \dots, N]$ e $y = [1, 2, \dots, N]$, sendo que f na posição (x, y) é regular ao nível de cinza (ou brilho) da imagem nesta localidade (GONZALEZ; WOODS, 2008).

Uma imagem digital é basicamente uma matriz bidimensional $M \times N$ (M representa o número de linhas e N o número de colunas), onde os índices das linhas e colunas resultam em um ponto na imagem, cujo valor corresponde à intensidade do nível de cinza

daquele ponto, que é chamado de pixel, que é abreviação de *picture elements* (elementos de imagem) (SANCHES, 2009). A Figura 2 ilustra uma representação de imagem digital.

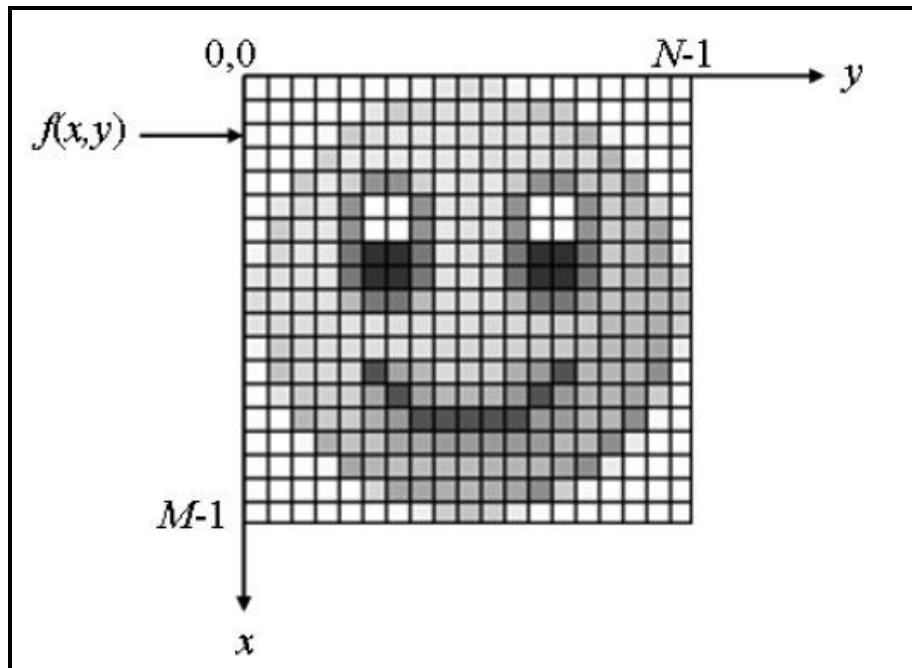


Figura 2: Representação de uma imagem digital (In: SANCHES, 2009)

2.1.2 Histograma

O histograma é uma forma de analisar e realçar a imagem. Ela fornece a informação de quantos pixels, na imagem, possuem um determinado nível de cinza (SANCHES, 2009). Esse nível pode ser definido entre 0 (preto) e 255 (branco), para uma imagem quantificada em 8 bits. O histograma não apresenta nenhuma informação espacial da imagem, mas indica a probabilidade de se encontrar um nível de cinza referente a um tom qualquer da imagem. Normalmente, tem-se no eixo horizontal a distribuição dos níveis de cinza e no eixo vertical a frequência em que ocorrem (MONTEIRO, 2005).

Pode ser estabelecida uma analogia entre o histograma de uma imagem e a função densidade de probabilidade, que pode ser vista como um modelo matemático da distribuição de tons de cinza de uma imagem. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu tom médio (se a imagem é predominantemente clara ou escura) (GONZALEZ; WOODS, 2008). A Figura 3 ilustra um histograma de níveis de cinza.

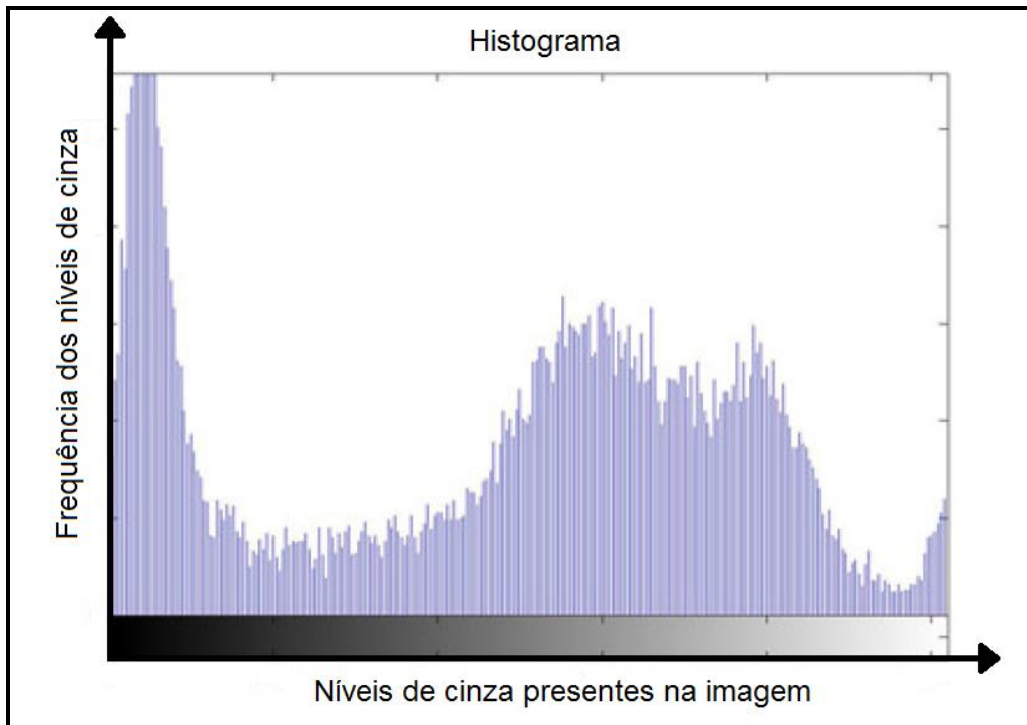


Figura 3: Ilustração do histograma de níveis de cinza de uma imagem (In: MONTEIRO, 2005)

A Figura 4 ilustra uma imagem com bom contraste e seu respectivo histograma.

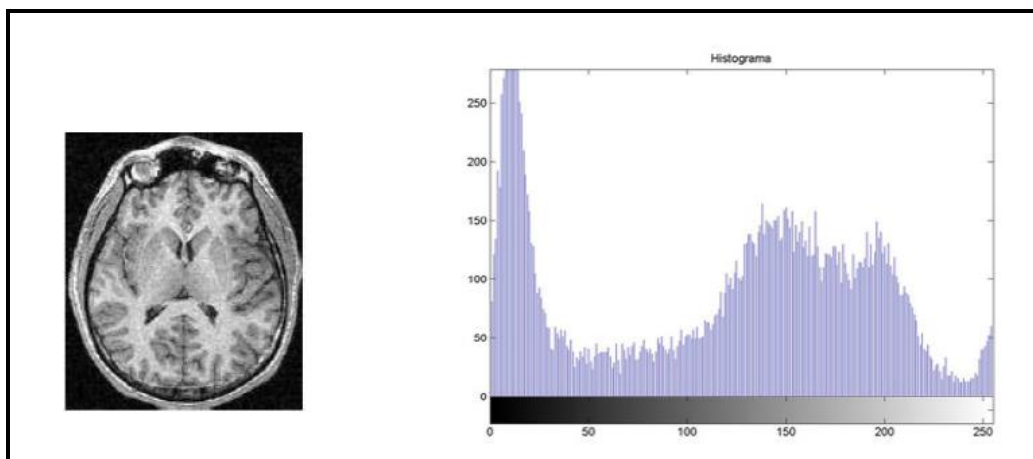


Figura 4: Imagem com bom Contraste e seu Respectivo Histograma (In: MONTEIRO, 2005)

2.1.3 Realce

O realce é em grande parte um processo subjetivo, baseado essencialmente em aspectos visualmente agradáveis para o espectador. Enquanto que a restauração se dá como um processo objetivo e visa recuperar imagens degradadas com uso de conhecimentos prévios do fenômeno de degradação. As técnicas de realce podem ser baseadas em processamento ponto a ponto, que altera o nível de cinza de um pixel independentemente

da natureza dos pixels adjacentes, ou pelo processo de filtragem, onde o novo valor depende dos valores dos pixels adjacentes (GONZALEZ; WOODS, 2008).

2.1.4 Contraste

Uma imagem com bom contraste possui um histograma com valores distribuídos ao longo de todo o intervalo da escala de tons. A manipulação do contraste de uma imagem tem como objetivo melhorar a sua qualidade visual ao olho humano sob critérios subjetivos. Considerando-se a iluminação insuficiente ou até mesmo um sensor defeituoso, o intervalo de intensidade a ser registrado nos pixels da imagem capturada pode ser reduzido, gerando uma imagem com pouco contraste e de difícil interpretação por homem ou máquina (MONTEIRO, 2005).

As técnicas de melhoria do contraste objetivam uma melhor qualidade visual, evidenciando as informações contidas na imagem. O processo não aumenta a quantidade de informação contida na imagem, porém torna mais fácil a sua percepção. Para aumentar o contraste, um mapeamento dos tons originais pode ser feito dentro do intervalo de tons de cinza [V_{min} - V_{max}] da imagem original para a variação máxima possível de visualização que, geralmente, é de (0 a 255). V_{min} e V_{max} são os valores de níveis de cinza mínimos e máximos, respectivamente, presentes na imagem original. Cada valor de nível de cinza na imagem original pode ser mapeado para um novo valor de acordo com uma função de transferência, que pode ser definida como: $y = f(x)$, onde x é o nível de cinza original e y é o novo valor correspondente (MONTEIRO, 2005). A Figura 5 apresenta uma imagem com baixo contraste e seu histograma.

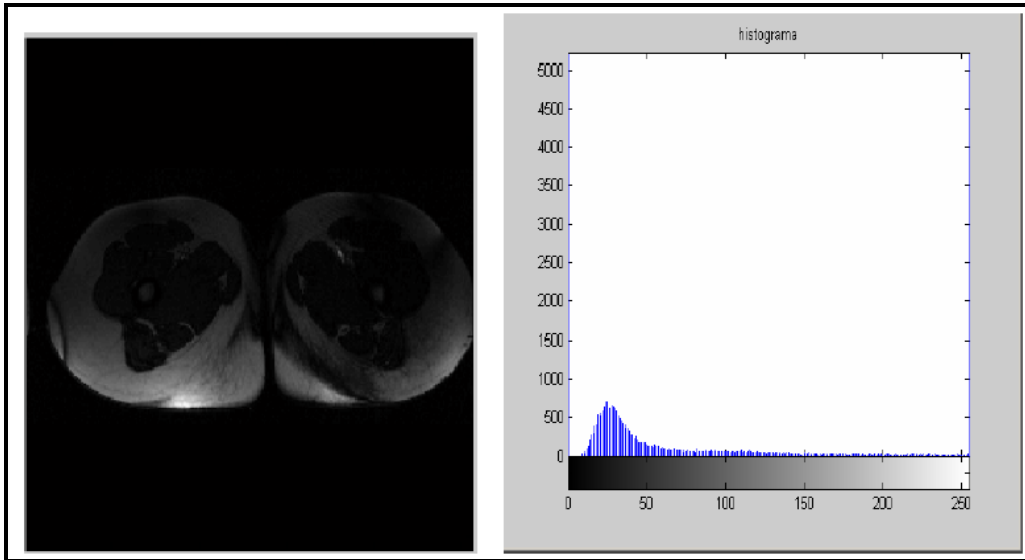


Figura 5: Imagem com baixo contraste e seu Histograma (In: MONTEIRO, 2005)

2.1.5 Equalização

É uma forma de manipulação de histograma que melhora automaticamente o contraste em áreas muito claras ou muito escuras presentes na imagem (GONZALEZ; WOODS, 2008). O processo de equalização de histograma tem por objetivo modificar o histograma da imagem original de tal forma que a imagem transformada tenha um histograma uniforme, ou seja, todos os níveis de cinza com o mesmo nível de intensidade devem aparecer na imagem com a mesma frequência (FARIA, 2010).

2.1.7 Limiarização (*Threshold*)

A técnica de *Thresholding* ou Limiarização é uma abordagem para a segmentação apoiada na análise da similaridade de níveis de cinza, de modo a extrair objetos de interesse mediante a definição de um limiar T que separa os agrupamentos de níveis de cinza da imagem. Por meio da análise do histograma da imagem, é possível estabelecer um valor para T na região situado entre picos que caracterizam regiões de interesse na imagem. Há diversas variantes de limiarização. A mais simples delas é a técnica do particionamento do histograma da imagem por um limiar único T . A segmentação se dá varrendo-se a imagem, pixel a pixel, e rotulando-se cada pixel como sendo do objeto ou do fundo, em função da relação entre o valor do pixel e o valor do limiar (MONTEIRO, 2005).

2.2 O PADRÃO DICOM

O protocolo DICOM constitui em uma padronização na comunicação e no armazenamento das imagens criadas por todos os equipamentos da área médica, de qualquer marca, em um formato eletrônico único e estruturado em protocolos. Esses protocolos permitem que as informações associadas às imagens geradas, possam ser trocadas entre si, obedecendo a uma série de regras.

Diferente dos outros formatos de imagens, ele permite que as informações dos pacientes sejam armazenadas juntamente com a imagem, mas de forma estruturada. Elas são armazenadas utilizando identificadores, conhecidos como “tags”, que identificam e limitam as informações. A imagem no padrão DICOM é baseada no formato JPEG, com ou sem compressão, dependendo do equipamento que a gerou (MONTEIRO, 2005).

Atualmente, o padrão DICOM encontra-se na terceira versão, porém segue em constante desenvolvimento e atualização. A principal função do comitê responsável pelo padrão é prezar pela compatibilidade eficaz com edições anteriores do DICOM e permitir a padronização de novos formatos de imagens diagnóstica.

2.2.1 Histórico

Na década de 1970, com a introdução da tomografia computadorizada (TC) e outras modalidades de diagnóstico por imagem digitais, o *American College Radiology* (ACR ou Colégio Americano de Radiologia) e a *National Electrical Manufacturers Association* (NEMA ou Associação Nacional de Fabricantes Elétricos) reconheceram a necessidade de um método padrão para a transferência de imagens e informações associadas entre dispositivos fabricados por diversos fornecedores (NEMA, 2016).

Em 1983, os membros do ACR e do NEMA formaram um comitê para iniciarem os trabalhos no desenvolvimento do padrão para promover a comunicação de informações contidas em uma imagem digital, seja qual fosse o fabricante do equipamento, para facilitar o desenvolvimento e expansão dos *Picture Archiving and Communication System* (PACS ou Sistema de Comunicação e Arquivamento de Imagens), permitir a comunicação com outros sistemas de informação hospitalar e permitir a criação de bases de dados de diagnóstico de informação (NEMA, 2016).

A primeira versão do padrão DICOM foi publicada somente em 1985, nomeada de ACR-NEMA 300-1985 ou “ACR-NEMA versão 1.0”, e a segunda versão em 1988, chamada de ACR-NEMA 300-1988 ou “ACR-NEMA versão 2.0”, semelhante a versão anterior, esta versão também aborda a transmissão de imagem ponto-a-ponto e fornece regras semânticas para a organização de mensagens. Contudo, com a disseminação da tecnologia de redes tornou-se necessário realizar mais algumas adaptações no padrão, pois o mesmo ainda possuía capacidade limitada para operar neste tipo de contexto (NEMA, 2016).

Em 1993, a terceira versão do padrão foi publicada e nomeada de ACR-NEMA DICOM ou “DICOM 3.0”, comumente denominado apenas como DICOM. Foram criados novos processos e principalmente o protocolo de comunicação para rede (NEMA, 2016).

O DICOM está essencialmente completo, mas em constantes revisões anuais e mudanças devido à evolução, pois ele é um padrão de muitas partes, podendo ter suas informações acrescidas quando necessário.

2.2.2 Partes do Padrão

A versão atual do DICOM consiste de 18 partes ativas e duas partes desativadas (NEMA, 2016):

Parte 1 - Introdução e visão geral: é o documento que contém um breve resumo do padrão. Este documento descreve a história, o escopo, objetivos e estrutura do padrão, e dá uma breve descrição de todas as outras partes.

Parte 2 - Conformidade: define os termos que as implementações devem aderir para ficar em conformidade com o padrão, seguindo os requisitos de conformidade e uma estrutura de declaração de conformidade, tanto a comunicação de rede como a troca de mídia.

Parte 3 - Definições de objetos de informações (IOD): descreve como os objetos de informação são definidos. Ela define as classes de objetos de informação (IOs) usadas no DICOM. São dois tipos de classes de objetos de informação, o normalizado que inclui somente os atributos inerentes à entidade real representada e o composto que pode ainda incluir atributos que estão relacionados, mas não inerentes à entidade do mundo real.

Parte 4 - Especificações da classe de serviço: Classes de serviço são construídas através de um conjunto de operações primitivas operando nas definições dos objetos de informação (IODs). Os serviços podem ser pensados como o desempenho das operações nos objetos de informação. Os papéis do SCU (*Service Class User* – Classe de Usuário de Serviço) e SCP (*Service Class Provider* - Classe de Provedor de Serviço) também são definidos nesta parte, o comportamento esperado para cada papel em cada classe de serviço é especificado. Isto permite a implementação e a compreensão dos usuários do que é esperado de um dispositivo que suporta uma classe particular de serviço.

Parte 5 - Estrutura de Dados e Codificação: define como as implementações DICOM devem construir e codificar a informação de conjuntos de dados resultantes da utilização de objetos de informação e classes de serviços. O suporte de um número de técnicas de compressão de imagem padrão (por exemplo, JPEG sem perdas e com perdas) é especificado. Aborda as regras de codificação necessários para construir um fluxo de dados a ser transportado em uma mensagem. Este fluxo de dados é produzido a partir de elementos de dados que compõem o conjunto de dados.

Parte 6 – Dicionário de Dados: é o registro que define a coleção de todos os elementos de dados disponíveis para representar a informação, juntamente com os elementos utilizados para a codificação de mídia e uma lista de itens identificados com exclusividade, que são os atributos. Para cada elemento é fornecido a sua etiqueta única, que consiste de um número de grupo e elemento, seu nome, sua representação (*string, integer, etc*), a multiplicidade (quantos valores por atributo) e o domínio de valores permitidos.

Parte 7 – Troca de informação: troca de informação em meio magnético usando o serviço de arquivo. Este serviço fornece três operações, tais como ler como FSR (*File-set Reader*) - Leitor de Conjunto de arquivos) ou escrever como FSC (*File-set Creator* - Criador de Conjunto de arquivos) e atualizar como FSU (*File-set Updater* – Atualizador de Conjunto de arquivo).

Parte 8 – Suporte à Comunicação de rede para troca de informação: define o suporte aos serviços de comunicação e os protocolos da camada superior necessárias em um ambiente de rede, para a troca de mensagens do DICOM. Atualmente o protocolo TCP/IP e qualquer outro que satisfaça o modelo de camadas ISO-OSI são suportados, mas a estrutura do padrão permite que novos protocolos sejam incorporados no futuro.

Parte 9 – Aposentado (Suporte à Comunicação de ponto-a-ponto para troca de informação): foi retirada do padrão DICOM. Anteriormente os serviços e protocolos eram utilizados nas comunicações ponto-a-ponto de uma maneira compatível com a ACR-NEMA 2.0.

Parte 10 – Armazenamento e formato de arquivo para intercâmbio de mídia: define um modelo geral para o armazenamento de informações de imagens médicas em mídia removível. O objetivo desta parte é o de proporcionar um quadro que permita o intercâmbio de vários tipos de imagens médicas e informações relacionadas em uma ampla gama de meios de armazenamento físico. Uma implementação de armazenamento de mídia pode reivindicar conformidade com o padrão. Essa conformidade é aplicável apenas à escrita, leitura e atualização do conteúdo da mídia de armazenamento.

Parte 11 - Perfis de aplicação de armazenamento de mídia: define subconjuntos específicos da aplicação do padrão DICOM a que uma execução pode verificar a versão implementada. Estes subconjuntos específicos da aplicação serão consultados para os perfis da aplicação nesta seção. Tal indicação de configuração aplica-se ao intercâmbio de imagens médicas e das informações relacionadas em meios de armazenamento para usos clínicos específicos.

Parte 12 – Formatos de mídia e mídia física para intercambio de mídia: facilitar o intercâmbio de informações entre aplicações em ambientes médicos, especificando uma estrutura para descrever a relação entre o modelo de armazenamento de mídia e um formato de mídia física e meios de comunicação específico.

Parte 13 – Aposentado (Gestão de Impressão ponto-a-ponto de Apoio à Comunicação): foi retirada do padrão DICOM. Definia os serviços e protocolos usados para comunicação ponto-a-ponto de serviços de gestão de impressão.

Parte 14 – Função de visualização padronizada de Tons de cinza: esta função proporciona métodos para calibrar um sistema de exibição especial para a finalidade de apresentar imagens de forma consistente em diferentes meios de visualização (por exemplo, monitores e impressoras).

Parte 15 – Segurança e perfis de Gestão do Sistema: consiste em referenciar os protocolos padrão desenvolvidos externamente, tais como DHCP, LDAP, TLS e ISCL. Protocolos de segurança podem usar técnicas de segurança como chaves públicas

e "cartões inteligentes". A encriptação de dados pode usar vários esquemas de criptografia de dados padronizados.

Parte 16 – Conteúdo de mapeamento de Recursos: define modelos de documentos estruturantes como o objeto de informação DICOM, conjuntos de termos codificados para uso em objetos de informação, um dicionário de termos definidos e mantidos por DICOM e traduções específicas do país em termos codificados.

Parte 17 – Informações Explicativas: define anexos de informações e normativas contendo informações explicativas.

Parte 18 – Web Services (Serviços Web): podem ser usados para recuperar ou armazenar um objeto DICOM. Os pedidos que recuperam dados especificam o tipo de mídia (formato) do corpo da resposta. Os pedidos que armazenam dados especificam o tipo de mídia do corpo da solicitação. As solicitações HTTP, tal como definidos no âmbito deste padrão são suficientes para o servidor HTTP para agir como um SCU DICOM (*Service Class User* – Classe de Usuário de Serviço) para recuperar ou armazenar os objetos solicitados de um SCP DICOM apropriado (*Service Class Provider* - Classe de Provedor de Serviço) usando a funcionalidade DICOM linha de base, o que significa dizer que o servidor HTTP pode atuar como um proxy para o DICOM SCP.

Parte 19 – Aplicação de Hospedagem: define uma API (*Application Programming Interface*) para um sistema de computação médica DICOM com base no qual os programas escritos para essa interface padronizada podem utilizar "plug-in". Um implementador *Hosting System* só precisa criar a API padronizada uma vez para suportar uma ampla variedade de add-on em aplicativos hospedados.

Parte 20 – Relatórios de imagem usando HL7(Health Level Seven International – Nível Sete de Saúde) CDA (Clinical Document Architecture - Documento de arquitetura clínica): define modelos para a codificação de relatórios de imagem utilizando o HL7 *Clinical Document Architecture* Release 2 (CDA R2, ou simplesmente CDA) Padrão. Neste âmbito são relatórios do procedimento clínicos para especialidades que usam imagens para diagnóstico ou fins terapêuticos. Um guia de implementação para o CDA, e que seja de acordo com a abordagem de modelos padronizados para guias de implementação CDA desenvolvidos pela HL7.

2.2.3 Conformidade (Aderência ao Padrão)

As implementações devem seguir dois princípios de conformidade para poder estarem de acordo com o padrão DICOM. Os princípios são os requisitos de conformidade e as declarações de conformidade. Os requisitos de conformidade especificam os requisitos gerais que devem ser cumpridos por qualquer implementação que alegue estar em conformidade. Enquanto isso, as declarações de conformidade definem a estrutura de uma implementação, especificando quais informações devem estar presentes em qualquer implementação com DICOM.

2.2.3.1 Declaração de Conformidade

Uma implementação não precisa empregar todos os componentes opcionais do padrão DICOM. Depois de se reunir os requisitos gerais mínimos, uma implementação DICOM pode utilizar qualquer uma das Classes *Service Object Pair* (SOP ou Serviços de Objetos Pareados), protocolos de comunicação, meios de armazenamento de aplicação, atributos opcionais, códigos e terminologia controlada, entre outros, necessários para realizar sua tarefa projetada.

A declaração de conformidade permite que o usuário determine quais componentes opcionais do padrão DICOM são suportados por uma implementação particular e quais extensões ou especializações adicionais estão inseridas na implementação. Portanto, ao comparar as demonstrações de conformidade de duas implementações diferentes, um usuário experiente deve ser capaz de determinar se é possível que as comunicações possam ser suportadas entre as duas implementações.

Diferentes estruturas são utilizadas pelo conteúdo das declarações de conformidade, conforme o caso da implementação possa suportar uma interface de rede DICOM, ou uma interface de armazenamento de mídias DICOM ou até uma combinação dos mesmos. Neste último caso, uma única declaração de conformidade deve conceder o que consiste em suas seções apropriadas.

2.2.3.1.1 Visão geral da seção de redes para declaração de conformidade

Uma implementação deve conter alguns elementos que estão contidos na própria declaração de conformidade para atender os requisitos de redes (NEMA, 2016):

- Uma visão geral funcional contendo o diagrama de fluxo de dados da aplicação que mostra todas as entidades da aplicação, incluindo quaisquer restrições de sequenciamento entre elas, mostrando como elas se relacionam com ambos, localmente e remotamente;
- Uma especificação mais detalhada de cada entidade de aplicação, listando as Classes SOP suportadas e visando as políticas com que iniciam ou aceitam associações entre estas;
- Para cada Entidade da aplicação e combinação de atividades do mundo real, uma descrição da proposta (por Associação Iniciação) e dos contextos de apresentação aceitáveis (por Associação Aceitação);
- Para cada Classe SOP relacionada à uma sintaxe abstrata, uma lista de todas as opções suportadas pelo SOP;
- Um conjunto de protocolos de comunicação que a implementação suporta;
- Uma descrição de qualquer extensões, especializações e privatizações divulgados publicamente na implementação;
- Uma seção descrevendo os detalhes de configuração relacionados ao DICOM;
- Uma descrição de todos os detalhes da implementação que podem ser relacionados com a conformidade do DICOM ou a interoperabilidade;
- Uma descrição de quais códigos e mecanismos de terminologia regulamentadas são utilizadas.

2.2.3.1.2 Visão geral da seção de armazenamento de mídias para declaração de conformidade

Uma implementação deve conter alguns elementos para atender os requisitos de armazenamento de mídias (NEMA, 2016):

- Semelhante à seção de redes, uma visão geral funcional contendo o diagrama de fluxo de dados da aplicação que mostra todas as entidades da aplicação, incluindo quaisquer restrições de sequenciamento entre elas, mostrando como elas se relacionam com ambos, localmente e remotamente;
- Uma especificação mais detalhada de cada entidade da aplicação, listando os perfis da aplicação de armazenamento de mídia suportadas (isto define as Classes SOP suportadas e as mídias selecionados), que descreve as políticas com as quais criam, ou atualizam os conjuntos de arquivos;
- Uma lista opcional de Classes SOP suportadas;
- Para cada Classe SOP de armazenamento de mídias relacionada a um perfil da aplicação de armazenamento de mídia, uma lista de todas as opções SOP suportadas;
- Para cada Classe SOP de armazenamento de médias relacionada a um perfil da aplicação de armazenamento de mídia, uma lista opcional de sintaxes de transferência suportadas;
- Uma descrição de qualquer extensões, especializações e privatizações divulgados publicamente na implementação;
- Uma seção descrevendo os detalhes de configuração relacionadas ao DICOM;
- Uma descrição de todos os detalhes da implementação que podem ser relacionados com a conformidade do DICOM ou a interoperabilidade;
- Uma descrição de quais códigos e mecanismos de terminologia regulamentadas são utilizadas.

2.2.3.2 Requisitos de conformidade

Para uma implementação estar de acordo com a conformidade do padrão é necessário que atenda pelo menos um dos dois requisitos essenciais de conformidade, o suporte a redes de computadores e o suporte ao armazenamento de mídias. Uma implementação também pode conter os dois requisitos.

2.2.3.2.1 Requisitos de conformidade em redes de computadores

Existem alguns fatores que auxiliam a satisfazer este requisito, dentre eles podem-se destacar a aceitação de um contexto de apresentação por parte dos dispositivos envolvidos, definindo papéis específicos para cada um, onde haverá papéis de cliente (SCU, *Service Class User* - Usuário de Classe de Serviço) e servidor (SCP, *Service Class Provider* - Provedor de Classe de Serviço). Também há a necessidade de uma produção e processamento de um Data Set (conjunto de dados), a obtenção do direito legítimo de um ID (Número Identificador) registrado por uma organização que seja condizente com a criação de UIDs (*Unique Identifiers* – Identificadores únicos), e obviamente suportar o protocolo de comunicação TCP/IP (NEMA, 2016).

2.2.3.2.2 Requisitos de conformidade em intercâmbios de mídias DICOM

O padrão também especifica métodos para proporcionar segurança em diferentes níveis do modelo de rede ISO/OSI, através da utilização de mecanismos específicos para uma determinada camada.

Para satisfazer os requisitos de conformidade do padrão, uma implementação deve possuir algumas atribuições como estar em conformidade com pelo menos um perfil de aplicação, como suportar umas das mídias físicas e formatos de mídias associadas, usar alguns dos termos utilizados pelo padrão para troca de dados tais como ler como FSR (*File-set Reader* - Leitor de Conjunto de arquivos) ou FSU (*File-set Updater* - Atualizador de Conjunto de arquivos) e escrever como FSC (*File-set Creator* - Criador de Conjunto de arquivos) ou FSU todas as Classes SOP definidas como obrigatórias por cada um dos perfis de aplicação suportados em uma das sintaxes de transferência obrigatórias, e por fim ser capaz de ignorar qualquer padrão ou Classe SOP privada que devem estar presentes no meio do armazenamento mas não estão definidos em qualquer um dos perfis de aplicação onde a conformidade é solicitada (RIBEIRO,2013).

2.2.4 Estrutura

2.2.4.1 Definições utilizadas pelo DICOM

Valor: Um componente de um campo de valor. Um campo de valor pode ser constituído por um ou mais destes componentes.

Campo de Valor: O campo dentro de um elemento de dados que contém o valor (s) desse elemento de dados.

Comprimento de Valor: O campo dentro de um elemento de dados que contém o comprimento do campo do valor do elemento de dados.

Valor de Multiplicidade (VM): Especifica o número de valores contidos no campo de valor de um elemento de dados.

Representação de Valor (VR): Especifica o tipo de dados e o formato do valor (s) contido no campo do valor de um elemento de dados.

Representação Campo de Valor: O campo onde a representação de valor de um elemento de dados é armazenada na codificação de uma estrutura de elemento de dados com VR explícito.

Comando: é um meio genérico para conduzir uma solicitação para operar sobre objetos de informação através de uma interface de rede.

Fluxo de Comandos: é o resultado de um conjunto de elementos de comandos DICOM usando o DICOM.

Big Endian: Uma forma de ordenação de bytes, onde valores binários (vários bytes) são codificados com o byte mais significativo codificado em primeiro lugar, e os bytes restantes codificados em ordem de importância decrescente.

Repertório de caracteres: um conjunto finito de caracteres diferentes que é considerado completo para um determinado propósito e é especificado independentemente da sua codificação (também referida como um conjunto de caracteres).

Elemento de dados: Uma unidade de informação, tal como definido por uma única entrada no dicionário de dados. Um atributo codificado de definição de objetos de informações (IOD), que é composto por, no mínimo, três campos: uma marca de elemento de dados, um comprimento de valor, e um campo de valor. Para algumas sintaxes de

transferência específicas, um elemento de dados também contém um campo *VR*, onde a representação do valor desse elemento de dados é especificada explicitamente.

Elemento de dados Tag: Um identificador exclusivo para um elemento de dados composto por um par ordenado de números (um número de grupo, seguido de um número de elementos).

Elemento de dados Type (Tipo): Usado para especificar se um atributo de um IOD ou um atributo de uma classe SOP é obrigatória ou opcional.

Conjunto de dados: são informações trocadas, e devem ser formadas por um conjunto estruturado de valores atribuídos diretamente ou indiretamente pelos objetos da informação relatada. O valor de cada atributo no conjunto de dados é expresso como um elemento de dados.

Prazo definido: O valor de um elemento de dados é um prazo definido quando o valor do elemento pode ser de um conjunto explicitamente especificado de valores padrão, e estes valores podem ser prorrogados por implementadores.

Número elemento: O segundo número do par ordenado de números que compõem uma marca de elemento de dados.

Enumerado Valor: O valor de um elemento de dados é um valor enumerado quando o valor do elemento deve ser de um conjunto explicitamente especificado de valores padrão, e esses valores não serão prorrogados por implementadores.

Número de Grupo: O primeiro número no par ordenado de números que compõem uma marca de elemento de dados.

Little Endian: Uma forma de ordenação de bytes, onde valores binários (vários bytes) são codificados com o byte menos significativo codificado em primeiro lugar; e os restantes bytes codificado em ordem de importância cada vez maior.

Célula de Pixel: O recipiente para um único valor de exemplo de Pixel, que pode incluir bits não utilizados. O tamanho de uma célula de pixel deve ser especificado pelos bits alocados (0028, 0100) Elemento de dados.

Dados de Pixel: dados gráficos (por exemplo, imagens) da variável de pixel de profundidade codificada nos dados de pixel, dados de pixel tipo *Float* ou *Double Float* Elemento de dados.

Pixel valor de exemplo: Um valor associado a um pixel individual. Um pixel individual consiste em um ou mais valores das amostras de pixel (por exemplo, imagens de cor).

Elemento de dados Privado: elemento de dados adicionais, definida por um implementador, para comunicar a informação que não está contida em elementos de dados padrão. Os elementos de dados privados têm números de grupo ímpares.

Repetindo Grupo: Elementos de dados normalizado dentro de um determinado intervalo de números de grupo, onde os elementos que têm números de elemento idênticos têm o mesmo significado dentro de cada grupo (e ao mesmo VR, VM, e Elemento de Dados *Type*). Repetindo Grupos só existirá para Curvas e Planos de Sobreposição (Números Grupo (50xx, eeee) e (60xx, eeee), respectivamente) e são um remanescente de versões deste antes da V3.0 de padrão.

Elemento de dados aposentado (não é mais utilizado): um elemento de dados que está começando a não ser suportado com a versão 3.0 do padrão. Implementações podem continuar a usar elementos de dados aposentado para efeitos de compatibilidade com versões anteriores da V3.0, mas não é uma exigência desta versão do padrão.

Sequência de Delimitação do Item: Item usado para marcar o fim de uma sequência de itens de comprimento indefinido. Este item é o último item em uma sequência de itens de comprimento indefinido.

Sequência de Itens (Valor Representação SQ): A Representação do Valor para elementos de dados que contém uma sequência de conjuntos de dados. Sequência de Itens permite conjuntos aninhados de dados.

Elemento de dados padrão: um elemento de dados definido no DICOM Padrão, portanto, listado no Dicionário de Dados do padrão.

Sintaxe e Transferência (padrão e privada): Um conjunto de regras de codificação que permitem a aplicação de entidades para negociar as técnicas de codificação (por exemplo, estrutura de elemento de dados, byte ordenação, compressão) que são capazes de suportar, permitindo assim que estas entidades de aplicação podem se comunicar.

Identificadores Únicos (UID): Uma sequência de caracteres que identifica exclusivamente uma grande variedade de itens, garantindo singularidade entre locais, fornecedores e equipamentos.

2.2.4.2 O arquivo DICOM

Os arquivos DICOM que estejam em conformidade com a parte 10 do padrão DICOM são geralmente referidos como "arquivos de formato DICOM" ou simplesmente "arquivos DICOM" e são representados como ".dcm" (NCBI, 2012). A Figura 6 apresenta o corpo de um arquivo DICOM.

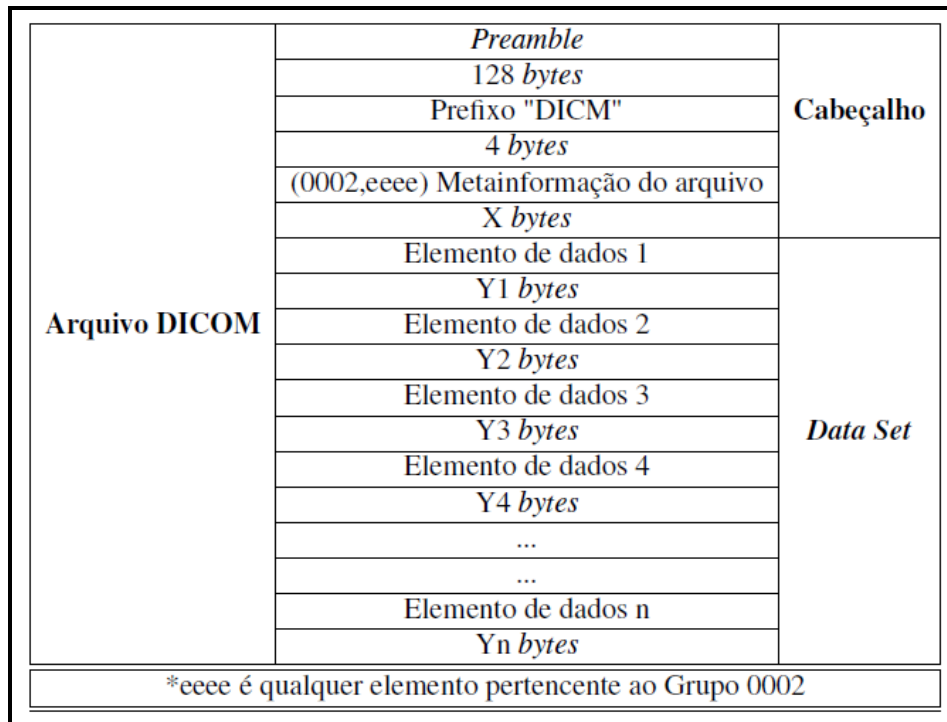


Figura 6: Representação do corpo de um arquivo DICOM (In: RIBEIRO, 2013)

2.2.4.2.1 Elemento de Dados

O arquivo DICOM inclui a identificação de informações sobre o conjunto de dados encapsulado. Este cabeçalho é constituído por um preâmbulo (*preamble*) de 128 bytes, que é a *string* e por padrão é preenchida com 0 bytes (espaços em branco) (RIBEIRO, 2013).

O preâmbulo do arquivo está disponível para uso, conforme definido pelos perfis de aplicação ou implementações específicas. Não é obrigado ser estruturado como um Elemento de dados DICOM com uma *Tag* e um comprimento. Destina-se a facilitar o acesso às imagens e outros dados no arquivo DICOM, fornecendo compatibilidade com uma série de computadores e formatos de imagem comumente usados (NEMA, 2016).

Após a preâmbulo, contém o prefixo "DICM" que ocupará 4 bytes no arquivo e possui os caracteres sempre em sua forma maiúscula. Tanto o prefixo "DICM" quanto a preâmbulo não estão estruturados como um elemento de dados DICOM, contendo uma *tag* e um comprimento definido (RIBEIRO, 2013).

O padrão DICOM aplica a seguinte definição para o seu conceito de *tag*: trata-se de um identificador único para uma informação de um elemento composto por um par ordenado de números (um número de Grupo seguido por um número de Elemento), dos quais são utilizados para identificar atributos e seus elementos de dados correspondentes (NEMA, 2016).

A partir do byte 133, em seguida do preâmbulo e do prefixo, pode-se encontrar a meta informação do arquivo DICOM, que carregará consigo todos os atributos que pertencem ao Grupo 0002, incluindo assim informações sobre a identificação dos Data Sets encapsulados (NEMA, 2016).

O elemento mais importante do Grupo 0002 é representado pela *tag* (0002,0010), este atributo recebe o nome de "*Transfer Syntax UID*" (Sintaxe de transferência com identificador único) e tem como função identificar o próximo Data Set. A Tabela 1 representa uma lista de outras possíveis *tags* (RIBEIRO,2013).

Depois do cabeçalho se iniciam os elementos de dados, sendo estes constituídos por quatro atributos, tais como a sua respectiva *tag* que conterà 32 bits, sendo 16 bits para o grupo da *tag* e 16 bits para o elemento de referido grupo, a sua *Value Representation* (VR ou Representação de Valor) que poderá ocupar 0, 16 ou 32 bits, dependendo do seu tipo, podendo ser uma data, um horário, um valor binário, ou uma *string*, ou ainda um nome, definida pelo Dicionário de dados, o seu *Value Length* (VL ou Valor de Comprimento) que poderá ser de 16 ou 32 bits, dependendo da codificação usada na Representação de Valor, e por fim o valor real do campo referente à *tag*, podendo ter um tamanho variável que pode chegar até mesmo a 1 *byte*.

A Tabela 1 apresenta algumas *tags* DICOM.

Informação Paciente	Tag DICOM (número grupo, número elemento)
ID Paciente	0010:0010
Nome Paciente	0010:0020
Data de nascimento Paciente	0010:0030
Sexo Paciente	0010:0040

Estudo de caso	Tag DICOM
ID Estudo	0020:0010
Nome do médico	0008:0090
Descrição Estudo	0008:1030
Data Estudo	0008:0020
Equipamento	Tag DICOM
Número de Série Equipamento	0020:0011
Modalidade Equipamento	0008:0060
Número de aquisição Equipamento	0020:0012
Imagem	Tag DICOM
Número de Instância	0020:0013
Altura Imagem	0028:0010
Largura Imagem	0028:0011

Tabela 1: Lista de tags DICOM (In: RIBEIRO, 2013)

A Figura 7 ilustra a representação dos elementos de dados.

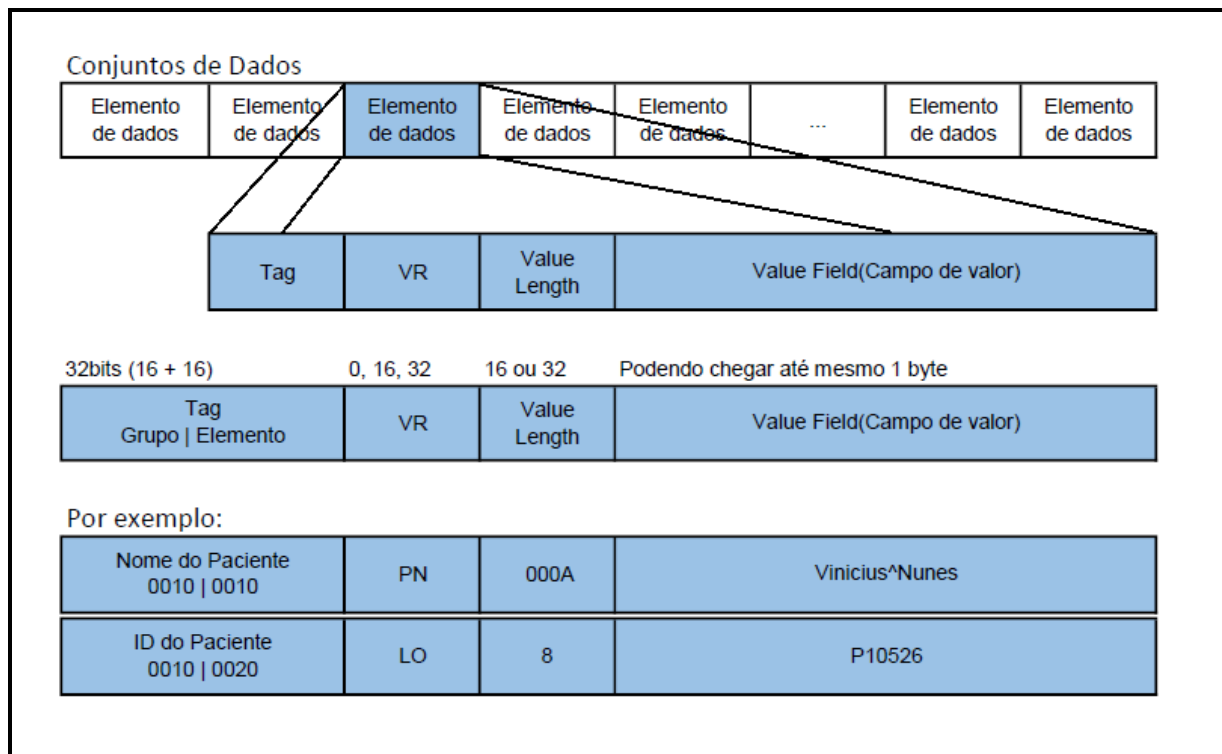


Figura 7: Representação dos elementos de dados (In: NEMA, 2016)

2.2.4.2.2 Cabeçalho de um arquivo DICOM

O tamanho do cabeçalho pode variar dependendo das informações armazenadas.

Abreviações utilizadas:

- SOP: *Service-object pair*
- UID: *Unique Identifier*

O campo “Tipo” indica o requisito de um objeto:

(1): requerido

(2): deve estar presente mas pode ter o tamanho zero se desconhecido.

(3): opcional

(1C) similar ao (1) mas é incluído se satisfizer uma condição.

(2C) similar ao (2) mas só pode ser incluído se satisfizer uma condição.

Exceto para o preâmbulo de 128 bytes e o prefixo de 4 bytes, a Meta informação do arquivo deve ser codificada utilizando a Sintaxe de Transferência VR *Little Endian* (UID = 1.2.840.10008.1.2.1). Os valores de cada Meta Elemento do arquivo devem ser preenchidos quando necessário para atingir um comprimento, pela sua representação de valor correspondente. A Representação valor desconhecido (UN) não devem ser utilizados na Meta informação do arquivo. Para compatibilidade com versões futuras da presente norma, qualquer *Tag* (0002, xxxx) não definida, devem ser ignoradas (NEMA,2016).

Valores de todas as *Tags* (0002, xxxx) são reservados para uso desta Norma e de versões posteriores do DICOM. Elementos de dados com um grupo de 0002 não devem ser usados em conjuntos de dados que não seja dentro da Meta informação do arquivo.

A Tabela 2 define o padrão DICOM, bem como as informações presentes no cabeçalho de um arquivo.

Nome do Atributo	Tag	Type	Descrição do Atributo
Preâmbulo do arquivo	<i>Nenhuma Tag ou comprimento de Campos</i>	1	Um campo de 128 bytes fixo disponível para Perfil de aplicação ou implementação específica. Se não for utilizado por um perfil de aplicação ou uma implementação específica, todos os bytes deve ser configurado para 00H. Leitores ou atualizadores não deve contar com o conteúdo deste preâmbulo para determinar que este arquivo é ou não é um arquivo DICOM.
DICOM Prefixo	<i>Nenhuma Tag ou comprimento de Campos</i>	1	Quatro bytes que contêm a cadeia de caracteres "DICM". Este prefixo se destina a ser usado para reconhecer que esse arquivo é ou não um arquivo DICOM.
Comprimento da Meta Informações do arquivo do grupo	(0002,0000)	1	Número de bytes seguindo esta Meta Elemento do arquivo (final do campo de Valor), incluindo informações do arquivo do grupo 2.
Versão Meta Informação do arquivo	(0002,0001)	1	Este é um campo de dois bytes, onde cada bit identifica uma versão deste cabeçalho de Meta Informações do arquivo na versão 1, o primeiro valor de byte é 00H e o segundo valor de byte é 01H. Implementações de leitura de arquivos com Meta informações, onde este atributo tem bit 0 (LSB) do segundo byte definido como 1 pode interpretar a Meta informações do arquivo especificadas. Todos os outros bits não devem ser verificados. Nota <i>Um campo de bits onde cada bit identifica uma versão, permite a indicação explícita do suporte de múltiplas versões anteriores. Futuras versões do arquivo de Meta informações que pode ser lido pela versão 1, leitores terão bit 0 do segundo byte definido como 1</i>
Armazenamento de mídia Classe SOP UID	(0002,0002)	1	Identifica a classe SOP associado ao conjunto de dados. Classe SOP UIDs permitidos para armazenamento de mídia.
Armazenamento de mídia Instância SOP UID	(0002,0003)	1	Identifica a Instância SOP associado com o conjunto de dados inseridos no arquivo e seguindo a Meta informações do arquivo.
Sintaxe de Transferência UID	(0002,0010)	1	Identifica a sintaxe de transferência usados para codificar o seguinte conjunto de dados. Esta sintaxe de transferência não se aplica a Meta informações do arquivo. Nota <i>Recomenda-se usar uma das sintaxes de transferência DICOM de suporte de codificação, Representação de valor explícito para facilitar a interpretação dos valores da Meta elemento do arquivo. Sintaxes de transferência JPIP Referenciado Pixel de dados não serão utilizados.</i>

Implementação da Classe UID	(0002,0012)	1	Identifica exclusivamente a implementação que escreveu este arquivo e seu conteúdo. Ele fornece uma identificação do tipo de aplicação que a última implementação escreveu o arquivo em caso de problemas de intercâmbio.
Nome da Versão da Implementação	(0002,0013)	3	Identifica uma versão para uma implementação de classe UID (0002,0012), utilizando até 16 caracteres.
Fonte do Título de Aplicação da Entidade - AE Title	(0002,0016)	3	<p>O Título de Aplicação da Entidade, que a implementação escreveu o conteúdo deste arquivo (última atualização). Se for utilizado, ele permite que o rastreamento da fonte de erros em caso de problemas de mídia de intercâmbio.</p> <p>Nota</p> <p><i>Se o conjunto de dados foi criado de novo pela aplicação ao gravar o arquivo, a sua AE título, se ele tiver um, pode ser usado. Se o conjunto de dados foram recebidos através da rede, existe um potencial de ambiguidade quanto ao fato do valor ser o mesmo que enviar o Título de Aplicação da Entidade (0002,0017) ou que recebe um Título de Aplicação da Entidade (0002,0018) ou algum outro valor.</i></p>
Enviando o Título de Aplicação de Entidade	(0002,0017)	3	<p>O Título de Aplicação da Entidade, que enviou o conteúdo deste arquivo em uma rede.</p> <p>Nota</p> <p><i>Este é o AE que era o remetente (fonte) do conteúdo (conjunto de dados), no caso de um conjunto de dados enviados através da rede (isto é, a chamada AET do SCU para uma operação de STORE C). Se o conjunto de dados foi criado, em vez de criado pela aplicação escrevendo o arquivo, ele não deve estar presente.</i></p>
Receber o Título da Entidade de Aplicação	(0002,0018)	3	<p>O Título de Aplicação da Entidade, que recebeu o conteúdo deste arquivo em uma rede.</p> <p>Nota</p> <p><i>Este é o AE que era o receptor (destino) do conteúdo (conjunto de dados), no caso de um conjunto de dados recebidos através da rede (isto é, o chamado AET da SCP para uma operação de STORE C). Se o conjunto de dados foi criado, em vez de novo pela aplicação escrevendo o arquivo, ele não deve estar presente.</i></p>
UID do Criador da informação privada	(0002,0100)	3	O UID do criador da informação privada (0002,0102).
Informação privada	(0002,0102)	1C	Contém informações particulares colocados na meta informações de arquivo. O criador deve ser identificado em (0002,0100). Necessário se o UID do Criador da Informação privada (0002,0100) está presente.

Tabela 2: Cabeçalho usado pelo padrão DICOM (In: NEMA, 2016)

2.2.4.2.3 Representação de Valor (VR)

A Representação de Valor (VR ou *Value Representation*) representa algum dos variados tipos e formatos de dados clínicos que o padrão DICOM suporta. Entre essa diversidade de formatos tem-se datas, horários, medidas de espaço, caracteres alfabéticos, entre outros, e com o intuito de poder representar toda essa diversidade o padrão DICOM definiu dentro de um conjunto de todas as VRs 31 tipos básicos de dados.

Cada VR possui a sua própria abreviação em duas letras, seguida de seu nome completo, possui também uma descrição do conteúdo da VR e de quais caracteres são permitidos nesses dados e por fim o comprimento do valor (RIBEIRO,2013).

A Tabela 3 apresenta algumas possíveis VRs do Padrão.

Nome da VR (completo e abreviado)	Definição do conteúdo da VR	Caracteres permitidos	Comprimento do valor
AE Entidade de Aplicação	Uma string de caracteres que identifica um nome do dispositivo	Caracteres maiúsculos, caracteres em branco (espaço) e o underline	16 bytes
AS Idade	Uma string de caracteres com um dos seguintes formatos - nnnD, nnnW, nnnM, nnnY; onde nnn deve conter o número de dias para D, semanas para W, meses para M, ou anos para Y.	"0" - "9", "D", "W", "M", "Y" do repertório de caracteres predefinido	4 bytes fixo
UI Identificador Único	Uma string de caracteres que contém um UID que é usada para identificar unicamente uma variedade de itens	"0" - "9", "." do repertório de caracteres padrão	64 bytes

Tabela 3: Representação de algumas das possíveis VRs do Padrão

2.3 DIAGNÓSTICO AUXILIADO POR COMPUTADOR (CAD)

O Diagnóstico auxiliado por computador ("*computer-aided diagnosis*" - CAD) pode ser definido como um diagnóstico feito por um radiologista que utiliza o resultado de análises de imagens radiográficas como uma "segunda opinião" para a tomada de decisões diagnósticas. É importante ressaltar que o computador é utilizado somente como uma

ferramenta para obtenção de informação adicional, sendo que o diagnóstico final sempre será feito pelo radiologista (DOI, 2001).

A finalidade do CAD é melhorar a precisão do diagnóstico, a interpretação da imagem radiológica, mediante o uso da resposta do computador como referência. Uma dupla leitura efetuada por dois especialistas pode aumentar a sensibilidade do diagnóstico. A proposta do CAD é funcionar como um segundo especialista.

Existem dois tipos de aplicações de sistemas CAD. O auxílio à detecção de lesões, com a localização de padrões anormais através de varredura da imagem pelo computador. E o outro é o auxílio ao diagnóstico, através da quantificação de características da imagem e sua classificação como correspondendo a padrões normais ou anormais. Em geral, os sistemas CAD utilizam-se de técnicas provenientes de duas áreas do conhecimento: visão computacional, que envolve o processamento de imagem para realce, segmentação e extração de atributos, e inteligência artificial, que inclui métodos para seleção de atributos e reconhecimento de padrões (AZEVEDO-MARQUES, 2001).

A Organização Mundial de Saúde (OMS) descreve a telemedicina como a oferta de serviços ligados aos cuidados com a saúde, nos casos em que a distância é um fator crítico. Tais serviços são providos por profissionais da área da saúde, usando tecnologias de informação e de comunicação para o intercâmbio de informações válidas para diagnósticos, prevenção e tratamento de doenças e a contínua educação de prestadores de serviços em saúde, assim como, para fins de pesquisas e avaliações, no interesse de melhorar a saúde das pessoas e de suas comunidades. A prática da telemedicina no Brasil é regulamentada pelas resoluções do Conselho Federal de Medicina números 1643/2002, 1638/2002 e 1639/2002. Além disso, há a Declaração sobre Responsabilidades e Normas Éticas na Utilização da Telemedicina aprovada na 51ª Assembleia Geral da Associação Médica Mundial em Tel Aviv (SEIXAS, 2005).

2.3.1 A Curva ROC

Existem vários índices de desempenho que podem ser utilizados na avaliação de sistemas de auxílio ao diagnóstico. Uma medida possível e muito utilizada é a análise da característica de resposta do observador (ROC).

No teste para detecção de uma doença, os resultados possíveis são positivos e negativos. Quando o resultado for positivo, o indivíduo pode ter a doença (verdadeiro positivo - VP) ou pode não a ter (falso positivo - FP). Da mesma forma, quando o resultado for negativo, o indivíduo pode não ter a doença (verdadeiro negativo - VN) como pode tê-la (falso negativo - FN) (PORTO, 2010).

A Tabela 4 apresenta os tipos de resultados usados em testes de detecção de doença.

TESTE (Resultado)	DOENÇA	
	Presente	Ausente
Positivo	Verdadeiro Positivo (VP)	Falso Positivo (FP)
Negativo	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Tabela 4: Representação Doença x Resultado (In: PORTO, 2010)

2.3.1.1 Sensibilidade e Especificidade

Para saber quão preciso é o teste, duas medidas foram criadas: a sensibilidade e a especificidade.

- Sensibilidade: mede a capacidade do teste em identificar corretamente a doença entre aqueles que a possuem, ou seja, o quão sensível é o teste.
- Especificidade: mede a capacidade do teste em excluir corretamente aqueles que não possuem a doença, ou seja, o quão específico o teste é.

A sensibilidade é a probabilidade de um teste dar positivo na presença da doença, sendo assim, avalia a capacidade do teste detectar a doença quando ela está presente:

$$\text{Sensibilidade} = \text{VP} / (\text{VP} + \text{FN}).$$

A especificidade é a probabilidade de um teste dar negativo na ausência da doença, sendo assim, avalia a capacidade do teste afastar a doença quando ela está ausente:

$$\text{Especificidade} = \text{VN} / (\text{VN} + \text{FP}).$$

A análise por curva ROC baseia-se na premissa de que o processo de decisão do observador pode ser modelado por uma variável randômica que se ajusta a um modelo bi-normal. As funções de densidade de probabilidade desta variável para as duas hipóteses (presença de anormalidade e ausência de anormalidade) são consideradas normalmente distribuídas. A menos que as condições do teste sejam perfeitas, estas distribuições se

sobrepõem e um limiar de decisão ("*threshold*") irá sempre envolver um compromisso entre os resultados falsos-positivos (não existe anormalidade, mas o observador indica que existe) e falsos-negativos (existe anormalidade, mas o observador indica que não existe) (AZEVEDO-MARQUES, 2001).

Uma mudança no limiar irá causar mudanças na especificidade e sensibilidade do teste. Os observadores humanos, assim como os sistemas de auxílio ao diagnóstico, operam em pontos ao longo da curva ROC. Os pontos de operação sobre a curva (sensibilidade vs. especificidade) mudam de acordo com os valores de custo atribuídos aos resultados falsos-positivos ou falsos-negativos (AZEVEDO-MARQUES, 2001). A Figura 8 apresenta um modelo de distribuição de probabilidade para análise.

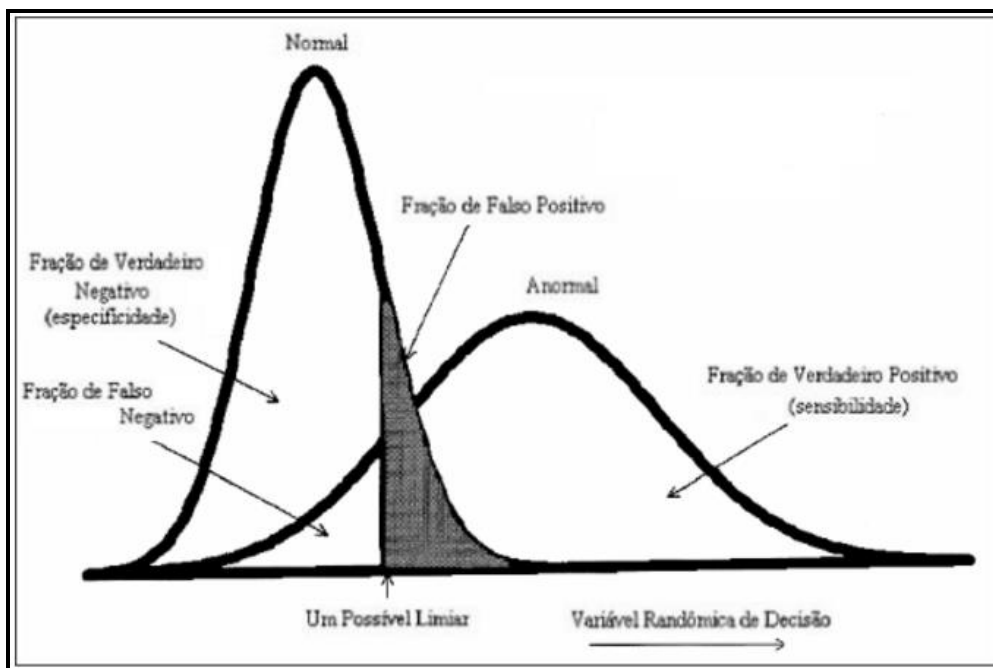


Figura 8: Modelo bi-normal de distribuição de probabilidade para análise por curva ROC (In: AZEVEDO-MARQUES, 2001)

Os parâmetros de aquisição da imagem digital, as características da base de dados utilizada para treinamento e teste dos algoritmos, a comparação entre métodos e a determinação da viabilidade de aplicação clínica são também características fundamentais no processo de avaliação de sistemas de CAD. Segundo Azevedo-Marques (2001), o uso de computadores para auxílio à análise de imagens radiológicas tem-se mostrado eficiente na melhoria da detecção e classificação de doenças.

2.3.2 Erro de diagnóstico radiológico e percepção

A negligência ocorre quando o "erro" infringe os princípios básicos da interpretação radiológica e/ou é causa substancial de prejuízo à saúde do paciente. O simples fato de uma lesão ser retrospectivamente visualizada também não significa necessariamente negligência (FENELON, 2003).

Alguns erros são cognitivos, isto é, resultam da má interpretação pela falta de conhecimento ou julgamento inadequado (interpretação incorreta). Existem também os erros causados pela utilização de técnica radiológica inadequada (exposição e posicionamento) (FENELON, 2003).

Os erros de percepção são aqueles em que as anormalidades radiológicas simplesmente não são detectadas pelo imaginologista em sua avaliação inicial. A falha na detecção geralmente é atribuída à sutileza do achado ou à sua má definição. Entretanto, o fenômeno da não visualização, de alterações "óbvias" e facilmente reconhecidas numa segunda análise, ainda não foi satisfatoriamente explicado (FENELON, 2003).

2.3.3 Exercício ilegal da radiologia e diagnóstico por imagem

O profissional sem diploma médico registrado no Conselho Regional de Medicina, que se apresenta como imaginologista, inclusive com registro no seu conselho de classe, pode ser denunciado criminalmente. Pois nenhum indivíduo pode apresentar-se como médico, sem ter cursado Medicina, em instituição de ensino superior reconhecida pelo Ministério da Educação (MEC). Utilizar o título de médico indevidamente é crime. A denúncia policial e no Conselho Regional de Medicina (CRM) pode ser apresentada por qualquer cidadão. Um profissional de nível superior que utilizar o título de médico, também poderá ser processado no seu próprio conselho de classe (Conselho Regional de Biomedicina, de Farmácia, de Enfermagem, etc.). Os procedimentos de diagnóstico e terapia são de exclusividade do médico (OLIVEIRA, 2012).

Laudos médicos são documentos particulares (propriedade do paciente). A falsificação desses documentos poderá caracterizar crime previsto no Código Penal Brasileiro, no Artigo 298: "Falsificar, no todo ou em parte, documento particular ou alterar documento

particular verdadeiro”. Pena: reclusão de 1 (um) a 5 (cinco) anos e multa (OLIVEIRA, 2012).

Ainda de acordo com o Código Penal Brasileiro, se o próprio paciente for o autor da falsificação, poderá ser processado por falsidade ideológica, de acordo com o Artigo 299: “Omitir, em documento público ou particular, declaração que dele devia constar, ou nele inserir ou fazer inserir declaração falsa ou diversa da que devia ser escrita, com o fim de prejudicar direito, criar obrigação ou alterar a verdade sobre fato juridicamente relevante. Diante de problema gerado por falsificação de laudo médico, deve a denúncia ser lavrada em delegacia de polícia e registrada em Boletim de Ocorrência (B.O)” (OLIVEIRA, 2012).

2.4 AS IMAGENS MÉDICAS

Imagens médicas têm por finalidade o auxílio no diagnóstico de anomalias e o fornecimento de material e informações para acompanhamento do paciente. Os sistemas CAD são sistemas computacionais com a finalidade de auxiliar o radiologista ou médico na tomada de decisão a respeito de um diagnóstico, obtido por resultados de uma análise computadorizada de imagens médicas (SAITO, 2007). A importância desses sistemas, é diminuir a taxa de erros de diagnósticos e mostrar que a utilização do CAD pode melhorar o desempenho de radiologistas no diagnóstico médico.

O processamento e armazenamento de imagens médicas vêm permitindo o aperfeiçoamento de diagnósticos e tratamentos de doenças de naturezas diversas. Essas imagens podem ser de diversos tipos de modalidades, como Radiografia, Ultra-Sonografia (US), Ressonância Magnética (MR), Tomografia Computadorizada (TC), entre outras.

2.4.1 Obtenção das imagens médicas

As imagens médicas envolvem diversas formas de manipulação, de visualização e de análise de estruturas corporais multidimensional. Os dados da imagem são obtidos através de um dispositivo de aquisição de imagens médicas. Existem dois métodos de obtenção de imagens médicas, os não invasivos e os invasivos (MONTEIRO, 2005).

2.4.1.1 Os não invasivos

- **Tomografia Computadorizada (CT):** originado do Raio-X, ela permite a obtenção de uma imagem de determinado segmento do corpo humano com a finalidade de visualização e estudos de suas estruturas anatômicas. Através de um tubo de raios-x (Rx) que gira em 360° em torno da região do corpo do paciente durante o exame, é obtido as “fatias” dessa região. As características dessa imagem obtida são o tamanho dos pixels, a matriz, o campo de visão, a escala de cinza e o janelamento. Cada pixel terá branco, preto ou escala de cinzas. O tamanho máximo que poderá ocupar na matriz é de 512 pixels em colunas e 512 pixels de linhas. A técnica de janelas (*windowing*) é utilizada para facilitar a visualização da imagem, uma variação de contrastes é denominada de janela. Pode-se ser gerado imagens 3D ou um conjunto de imagens de “fatias” 2D (MONTEIRO, 2005).
- **Ressonância Magnética (MR):** utiliza sinais provenientes do núcleo de hidrogênio contidos nos componentes do corpo que contém grande quantidade de hidrogênio que são a água e gordura. As imagens são obtidas por sinais de frequência de ondas de rádio. Não utiliza radiação e as imagens produzidas são de boa qualidade, uma desvantagem é não poder utilizá-la para fazer exames de ossos e órgãos que movimentam como o pulmão, coração e outros. As imagens geradas são sempre “fatias” 2D e adquiridas em planos: axial, sagital, coronal ou oblíquo (MONTEIRO, 2005).

Axial: Delimita o plano de corte numa vista superior e inferior, paralelo ao solo.

Sagital: Delimita o plano de corte numa vista direita e esquerda. Reformatado sempre da esquerda para a direita, perpendicular ao solo.

Coronal: Delimita o plano de corte numa vista posterior – anterior (ou ainda frente - costas) pode ser descrito como sendo um plano perpendicular ao solo, porém, ortogonal ao plano sagital.

Oblíquo: pode ter qualquer orientação.

A Figura 9 apresenta os tipos de planos de aquisição de imagens.

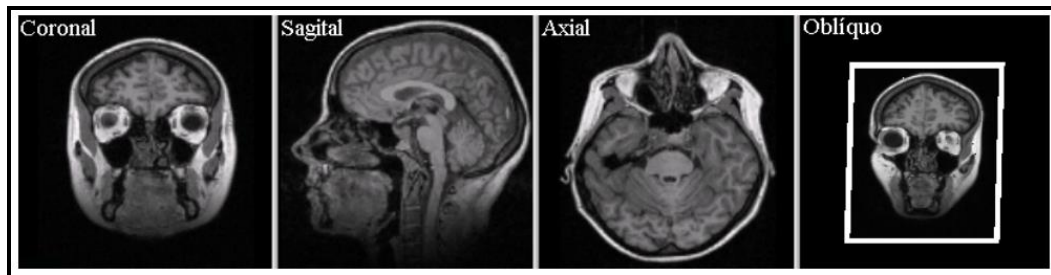


Figura 9: Tipos de Planos de Aquisição de Imagens (In: MANSSOUR; FREITAS, 2002)

- **Ultra-som (US):** As imagens obtidas são produzidas por cristais que transformam energia elétrica em energia acústica ou o inverso. Os cristais vibram com frequências de cerca de 2 a 10 MHz. Através das velocidades de transmissão das ondas sonoras, são obtidas as imagens, a partir da interação do som e tecidos. É utilizado um transdutor (Sonda) que converte uma energia em outra.

2.4.1.2 Os invasivos

Um método invasivo é aquele que exige a penetração física nos tecidos. Existem níveis diferentes de invasão que podem ser considerados métodos semi-invasivos.

Pode-se citar como exemplo o Cateterismo Cardíaco ou Cinecoronariografia, que é o estudo das artérias do coração (coronárias). Neste método é introduzido um tubo fino e flexível chamado de cateter em uma veia ou artéria, guiado até o coração para a visualização. A Angiografia Digital que é o estudo dos vasos sanguíneos: através de uma injeção de contraste e utilizando radiação é possível detecção de anomalias. Existe também a Endoscopia que possibilita a visualização no interior de alguns órgãos, através de instrumentos óticos que podem ou não ser adaptados com câmeras e monitores.

2.5 TECNOLOGIAS UTILIZADAS

Linguagem C++: Segundo Stroustrup (2016), seu criador, a linguagem C++ pode ser definida como uma linguagem para a programação de sistemas que é uma versão do C melhorada, oferece suporte a abstração de dados, suporta programação orientada a objetos e suporta programação genérica.

A linguagem C++, apesar de ter um criador, não possui um proprietário, ela é regulamentada pelas normas ISO/IEC 14882 “The C++ Standard”. As normas são definidas pelo comitê de gestão ISO/IEC JTC1/SC22 que define um padrão a cada 3 anos, visando aprimorar a linguagem conforme as necessidades. No momento de escrita deste trabalho os dois últimos padrões são o C++11 e C++14 respectivamente. Para o desenvolvimento do visualizador será utilizado o padrão C++11 (STROUSTRUP, 2016).

Ferramenta ITK: ITK (*Insight Segmentation and Registration Toolkit*) é uma biblioteca multi-plataforma de código aberto, escrita em C++, que fornece aos desenvolvedores com um extenso conjunto de ferramentas de software para análise de imagem, processamento de imagem, registro de imagem e segmentação de imagem. A Kitware e mais 5 colaboradores, a GE Corporate R&D, MathSoft, a Universidade da Carolina do Norte, a Universidade de Tennessee e a Universidade da Pensilvânia construíram a ferramenta (ITK, 2016).

Ferramenta VTK: VTK (*Visualization Toolkit*) é uma biblioteca multi-plataforma, de código aberto, escrita em C++, que fornece aos desenvolvedores ferramentas para computação gráfica 3D, processamento de imagem e visualização. Desenvolvida inicialmente em 1993, pela *Kitware* (VTK, 2016).

Framework Qt: um kit de desenvolvimento gráfico multi-plataforma para o desenvolvimento de aplicações. Foi criado em 1991, pela *Trolltech* e atualmente é propriedade da Digia Plc. Sua arquitetura pode ser dividida em módulos que correspondem à sua biblioteca e podem ser ativados ou desativados em tempo de compilação, permitindo ajustes no conjunto de recursos, fazendo com que todos os módulos se encaixem (THELIN, 2007).

Microsoft Visual Studio 2013: O Microsoft Visual Studio é um pacote de programas da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# e J#. Também é um grande produto de desenvolvimento na área web, usando a plataforma do ASP.NET (MSDN, 2016).

3 – METODOLOGIA

Primeiramente será utilizado o ITK para obtenção e manipulação das informações e dados médicos do padrão DICOM. O VTK será utilizado para a visualização das imagens na tela. A biblioteca não possui suporte para interface gráfica. Toda codificação será feita com a *framework* Qt com o compilador MVSC 2013, utilizando a linguagem C++.

Para criação de uma interface gráfica agradável para a implementação, será utilizado a *framework* Qt. Mesmo o VTK não suportando uma interface gráfica, ele possui um plug-in, o QVTKWidget, que serve para uma comunicação com o Qt. Esse plug-in possibilita a visualização, na GUI do Qt, das imagens codificadas utilizando o VTK.

3.1 LISTA DE EVENTOS

O visualizador tem a seguinte lista de eventos:

1. Abrir Imagem
2. Visualizar Imagem
3. Selecionar fatias
4. Exibir *Tags*
5. Aplicar zoom
6. Aplicar Contraste
7. Aplicar Rotação
8. Aplicar Medida
9. Gerar Laudo
10. Exportar Imagem
11. Imprimir Imagem

3.2 DIAGRAMA DE CASOS DE USO

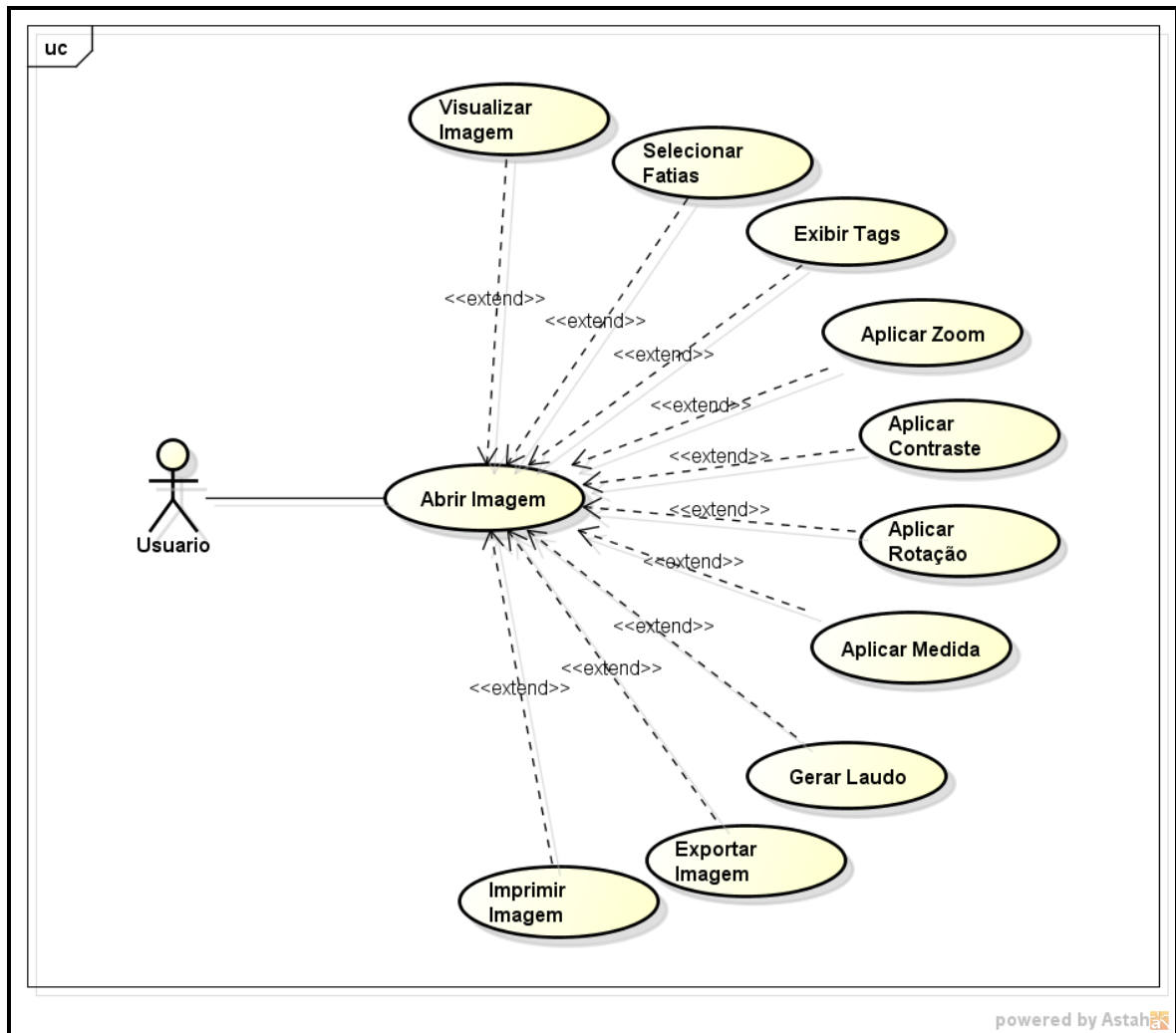


Figura 10: Diagrama de Casos de Uso

3.3 ESPECIFICAÇÃO DE CASOS DE USO

3.3.1 Abrir Imagem

Nome do Caso de Uso:	Abrir Imagem
Atores:	Usuário
Pré-condições:	Possuir as imagens para estudo
Fluxo Principal:	<p>1 – O usuário abre o programa</p> <p>2 – O sistema dispõe a opção “Abrir Imagem”</p> <p>3 – O usuário seleciona a Opção</p>

	4 – O usuário seleciona a imagem desejada
Fluxo Alternativos:	O usuário fecha o programa
Fluxo de Exceção:	

Tabela 5: Abrir Imagem

3.3.2 Visualizar Imagem

Nome do Caso de Uso:	Abrir Imagem
Atores:	Usuário
Pré-condições:	Ter selecionado uma imagem para estudo
Fluxo Principal:	1 – O usuário visualiza a tela principal, que já é a tela de visualização 2 – O sistema dispõe funcionalidades na barra de menu
Fluxo Alternativos:	Abrir outra imagem
Fluxo de Exceção:	

Tabela 6: Visualizar Imagem

3.3.3 Selecionar fatias

Nome do Caso de Uso:	Selecionar fatias
Atores:	Usuário
Pré-condições:	A imagem de estudo deve ser de CT ou MR
Fluxo Principal:	1 – O sistema dispõe uma lista de todas as fatias da imagem 2 – O usuário seleciona a fatia desejada por meio de um slider, ou utiliza as setas do teclado ou scroll do mouse para troca de fatia
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 7: Selecionar fatias

3.3.4 Exibir Tags

Nome do Caso de Uso:	Exibir Tags
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu tags</p> <p>2 – O usuário seleciona a opção tags</p> <p>3 – O sistema dispõe uma janela contendo uma tabela com as tags</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 8: Exibir tags

3.3.5 Aplicar zoom

Nome do Caso de Uso:	Aplicar zoom
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu</p> <p>2 – O usuário clica na opção zoom in ou zoom out</p> <p>3 – O usuário pode utilizar botão direito do mouse pressionado e mover o cursor</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 9: Aplicar zoom

3.3.6 Aplicar Contraste

Nome do Caso de Uso:	Aplicar Contraste
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu</p> <p>2 – O usuário seleciona a opção Contraste</p> <p>3 – O usuário pode utilizar botão esquerdo do mouse pressionado e mover o cursor</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 10: Aplicar Contraste

3.3.7 Aplicar Rotação

Nome do Caso de Uso:	Aplicar Rotação
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu</p> <p>2 – O usuário clica na opção Rotação</p> <p>3 – O usuário pode utilizar o spinbox ou um slider para selecionar o grau da rotação.</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 11: Aplicar Rotação

3.3.8 Aplicar Medida

Nome do Caso de Uso:	Aplicar Medida
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	1 – O sistema dispõe uma barra de menu 2 – O usuário seleciona a opção Medida 3 – O usuário pode utilizar botão esquerdo do mouse pressionado e mover o cursor
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 12: Aplicar Medida

3.3.9 Exportar imagem

Nome do Caso de Uso:	Exportar imagem
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	1 – O sistema dispõe uma barra de menu 2 – O usuário seleciona a opção “Exportar” 3 – O usuário pode salvar as alterações em um arquivo novo
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 13: Exportar Imagem

3.3.10 Gerar laudo

Nome do Caso de Uso:	Gerar laudo
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu</p> <p>2 – O usuário seleciona a opção “Gerar laudo”</p> <p>3 – O sistema dispõe uma janela de texto</p> <p>4 – O usuário digita o laudo do paciente</p> <p>5 – O usuário salva o laudo em .txt</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 14: Gerar Laudo

3.3.11 Imprimir Imagem

Nome do Caso de Uso:	Imprimir Imagem
Atores:	Usuário
Pré-condições:	Imagem na tela de visualização
Fluxo Principal:	<p>1 – O sistema dispõe uma barra de menu</p> <p>2 – O usuário seleciona as imagens desejadas</p> <p>3 – O sistema dispõe uma opção “imprimir imagens”</p> <p>4 – O usuário seleciona a impressora</p> <p>5 – O usuário imprime as imagens</p>
Fluxo Alternativos:	
Fluxo de Exceção:	

Tabela 15: Imprimir Imagem

3.4 DIAGRAMA DE ATIVIDADES

3.4.1 Abrir Imagem

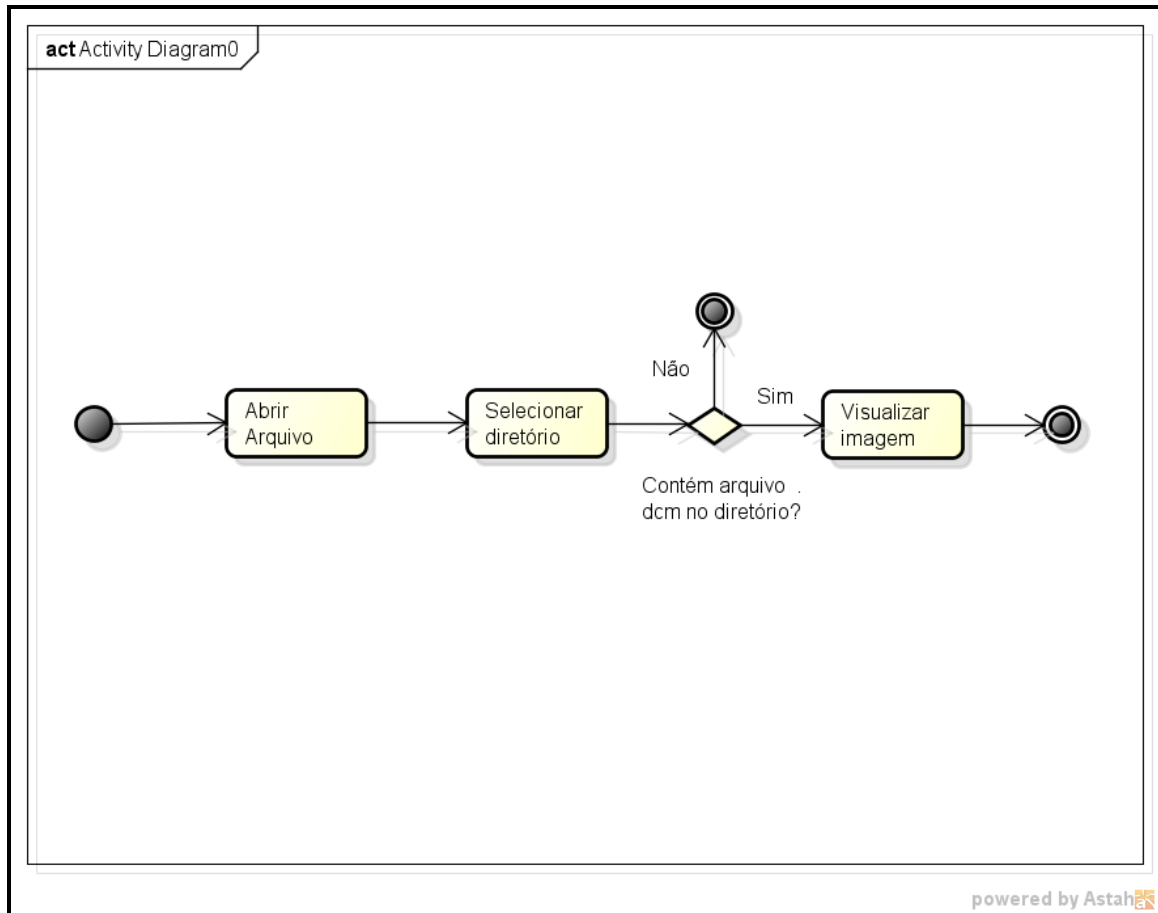


Figura 11: Diagrama de Atividades Abrir Imagem

3.4.2 Exibir Tags

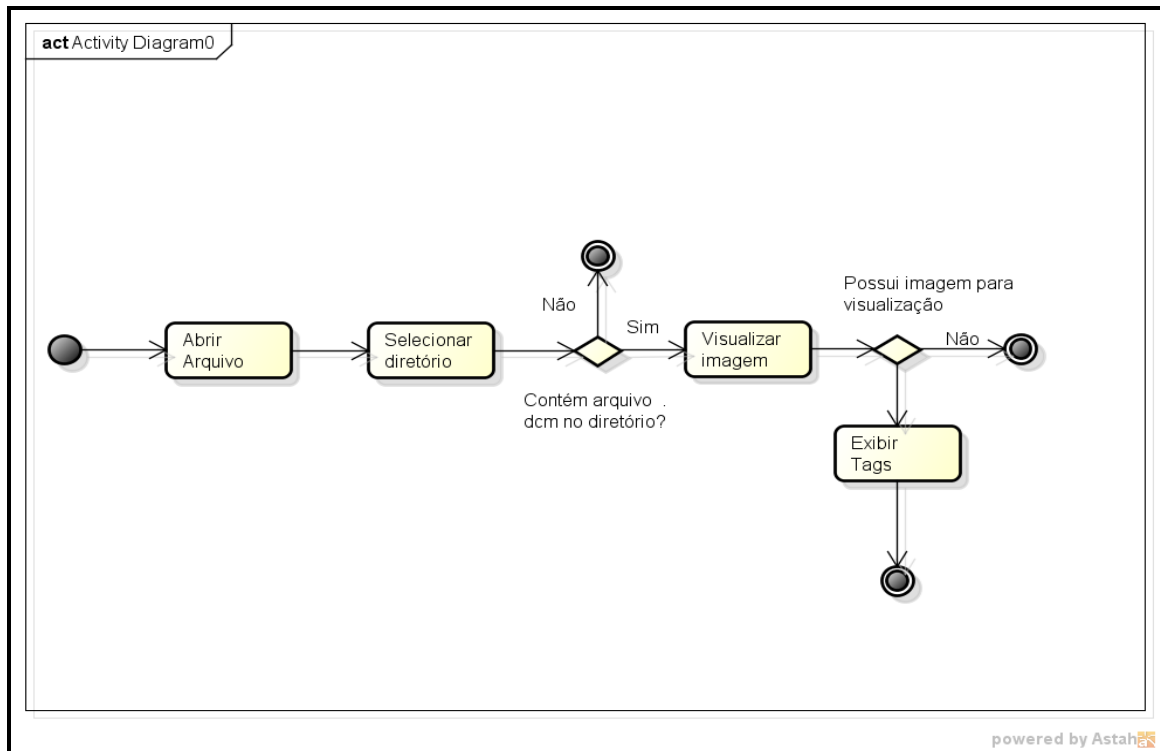


Figura 12: Diagrama de Atividades Exibir Tags

3.4.3 Aplicar zoom

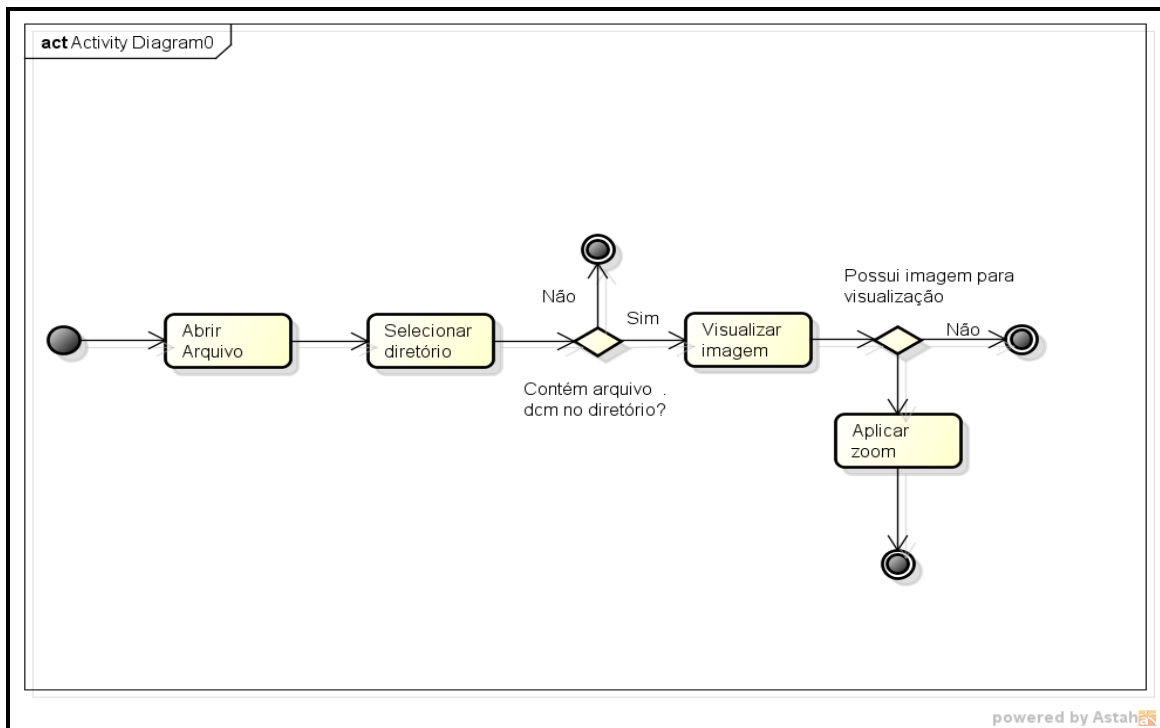


Figura 13: Diagrama de Atividades Aplicar zoom

3.4.4 Aplicar Contraste

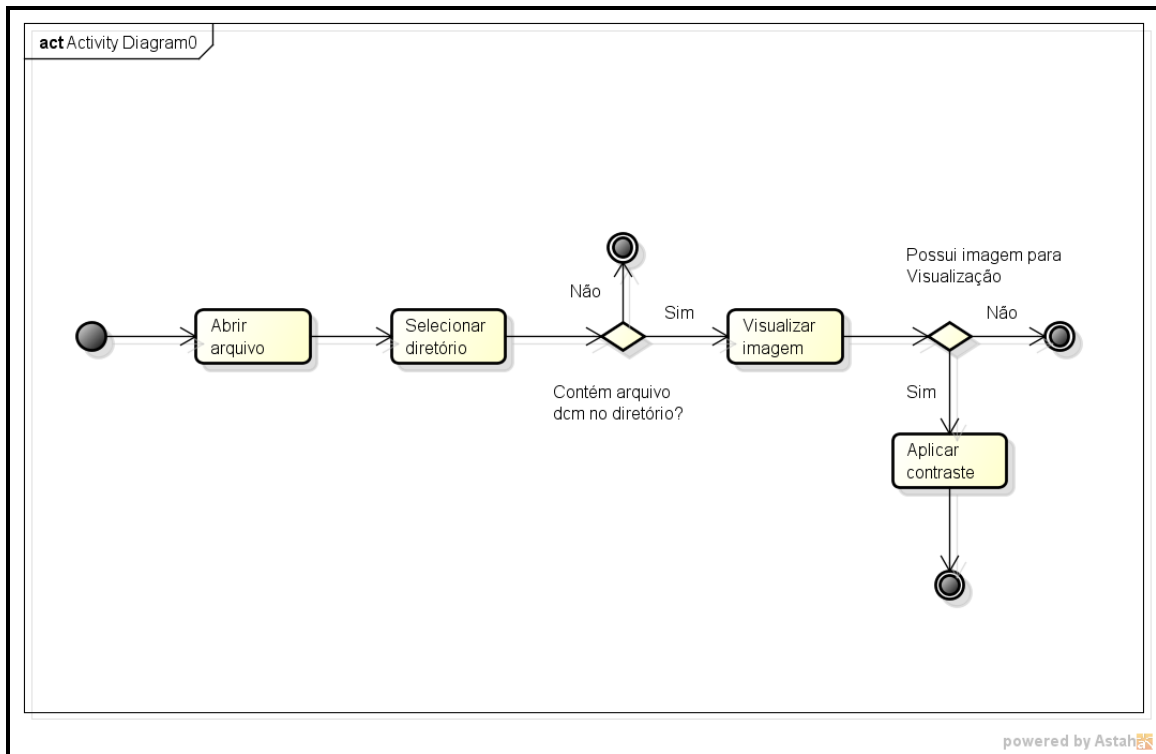


Figura 14: Diagrama de Atividades Aplicar Contraste

3.4.5 Aplicar Rotação

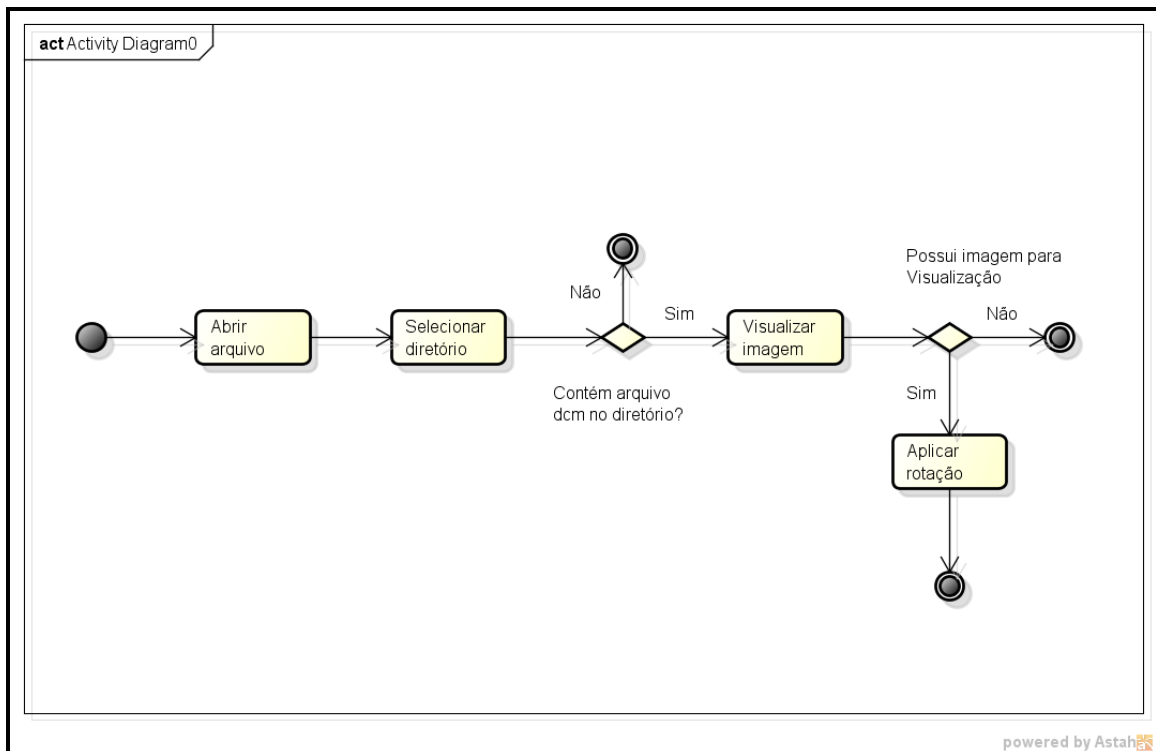


Figura 15: Diagrama de Atividades Aplicar Rotação

3.4.6 Aplicar Medida

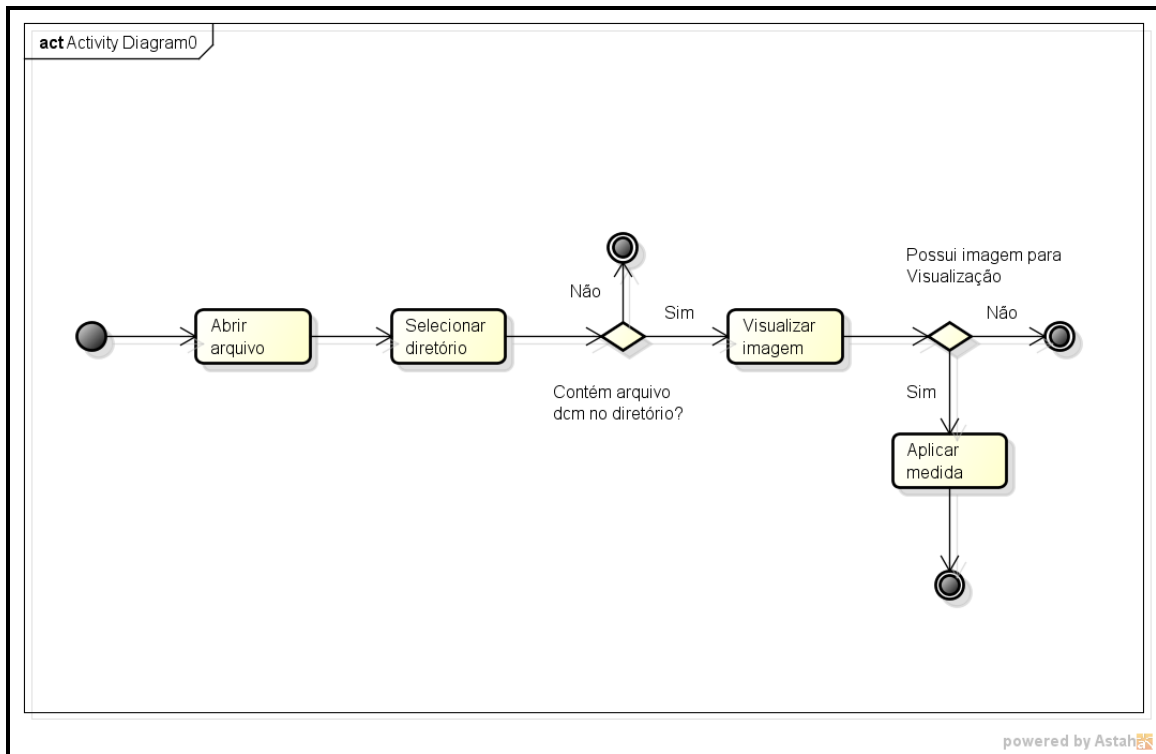


Figura 16: Diagrama de Atividades Aplicar Medida

3.4.7 Exportar imagem

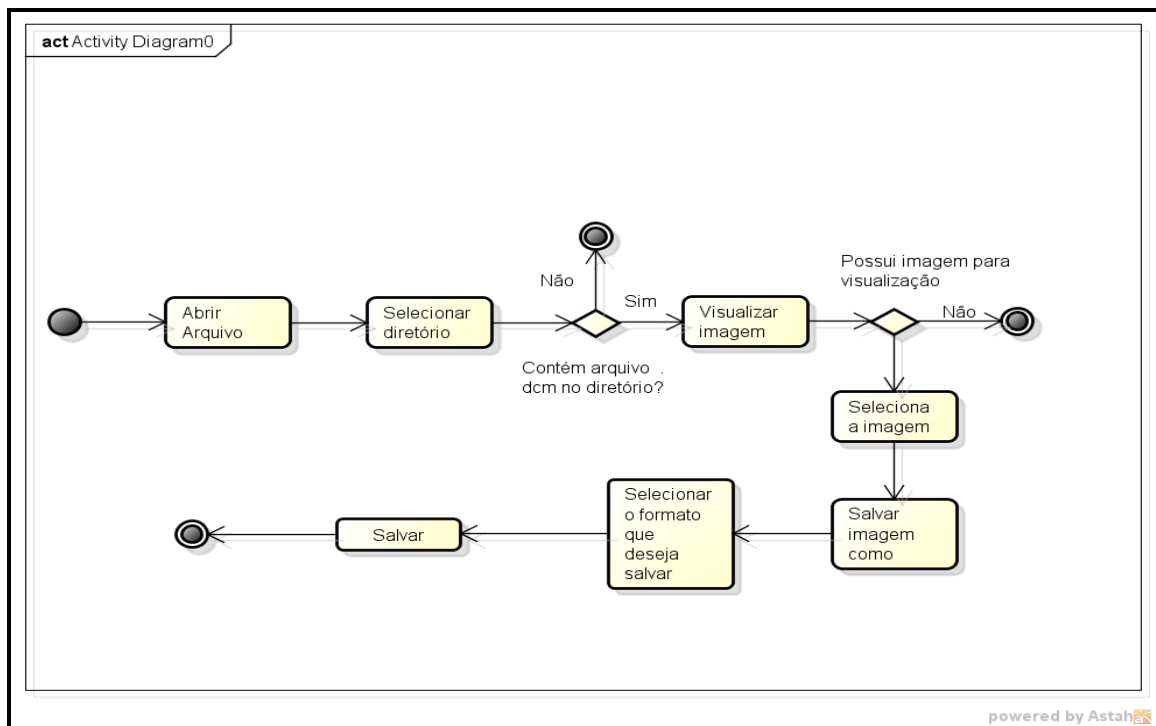


Figura 17: Diagrama de Atividades Exportar Imagem

3.4.8 Imprimir Imagem

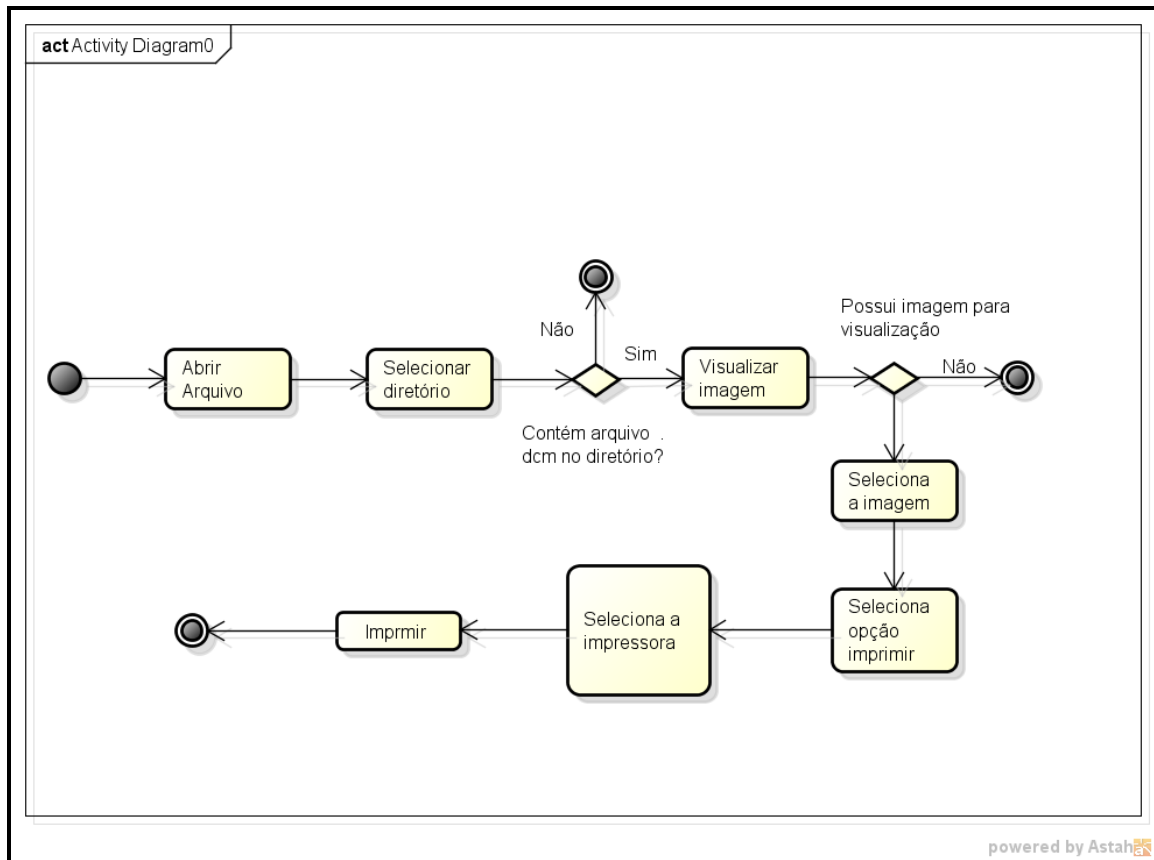


Figura 18: Diagrama de Atividades Imprimir Imagem

3.5 ESTRUTURA ANALÍTICA DO PROJETO (EAP)

Estrutura Analítica do Projeto (EAP) é um recurso que tem o objetivo de dividir o projeto em partes menores, assim facilitando o entendimento e gerenciamento de todos os processos envolvidos do projeto. A Figura 19 ilustra a EAP definida para o presente projeto.

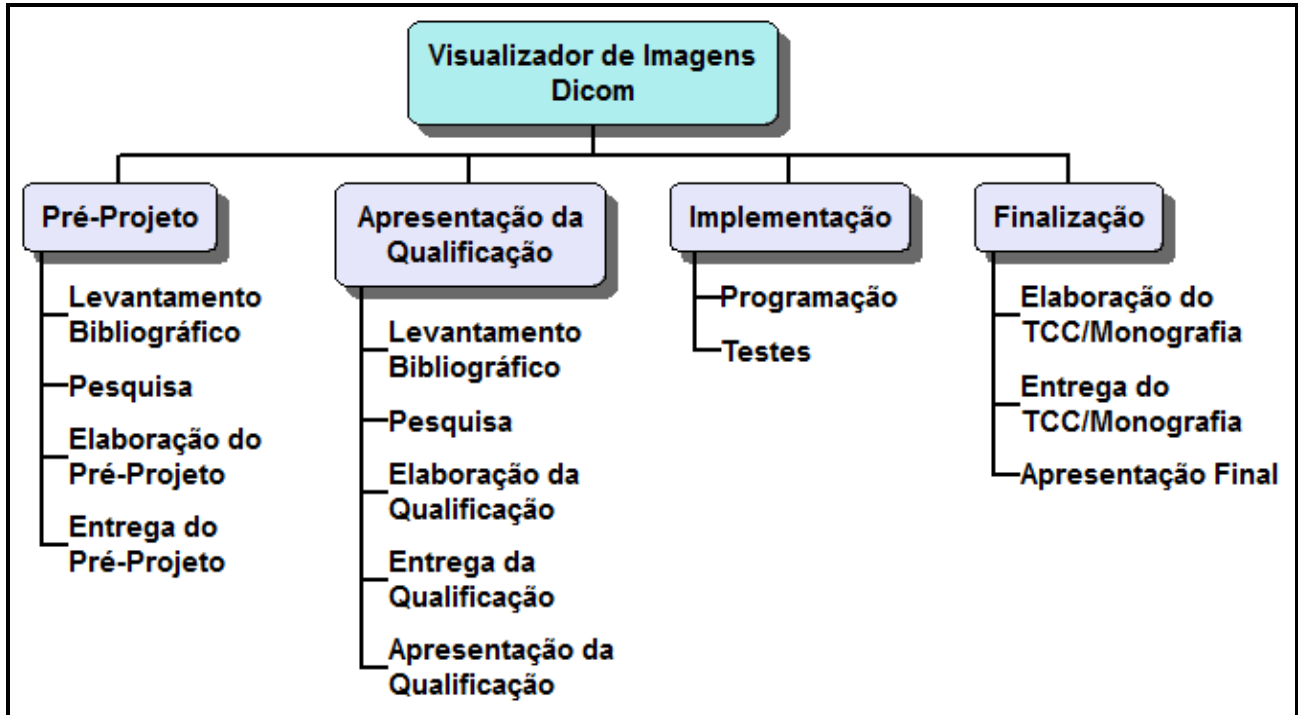


Figura 19: EAP, representação dos processos envolvidos no projeto

4 – DESENVOLVIMENTO

4.1 CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento foi utilizado o compilador do Microsoft Visual Studio 2013. Dessa forma, o Qt adotado é o baseado no compilador do MSVC2013 e OpenGL.

Após a instalação do Qt, devem ser criadas as variáveis de ambiente responsáveis pela configuração:

- QTDIR: "C:\Qt\Qt5.5.1\5.5\msvc2013"
- Adicionado "%QTDIR%\bin" na variável de ambiente PATH

Para a configuração das Bibliotecas VTK e ITK foi necessário utilizar a ferramenta CMake. O CMake é *open-source* e foi projetado para construir, testar e empacotar software. É usado para controlar o processo de compilação de software utilizando uma plataforma simples e arquivos de configuração do compilador independente. Foi criado pela Kitware em resposta à necessidade de um ambiente de compilação multi-plataforma para projetos de código aberto, como ITK e VTK (CMake, 2016).

Após baixar os arquivos VTK, deve-se criar uma pasta para receber os arquivos gerados pelo CMake. A pasta "*build*" é o diretório de destino para os arquivos gerados. A pasta do VTK é o diretório de origem que contém o arquivo "CMakeLists.txt". Com as pastas de origem e destino preenchidas no CMake, o botão "*Configure*" deve ser pressionado para exibir as opções de configurações. Pressionando o botão "*Configure*", uma janela de seleção de compilador será exibida e a opção 'MSVC2013 deve ser selecionada. Na Figura 18 é ilustrado o procedimento para a geração de arquivos com CMake:

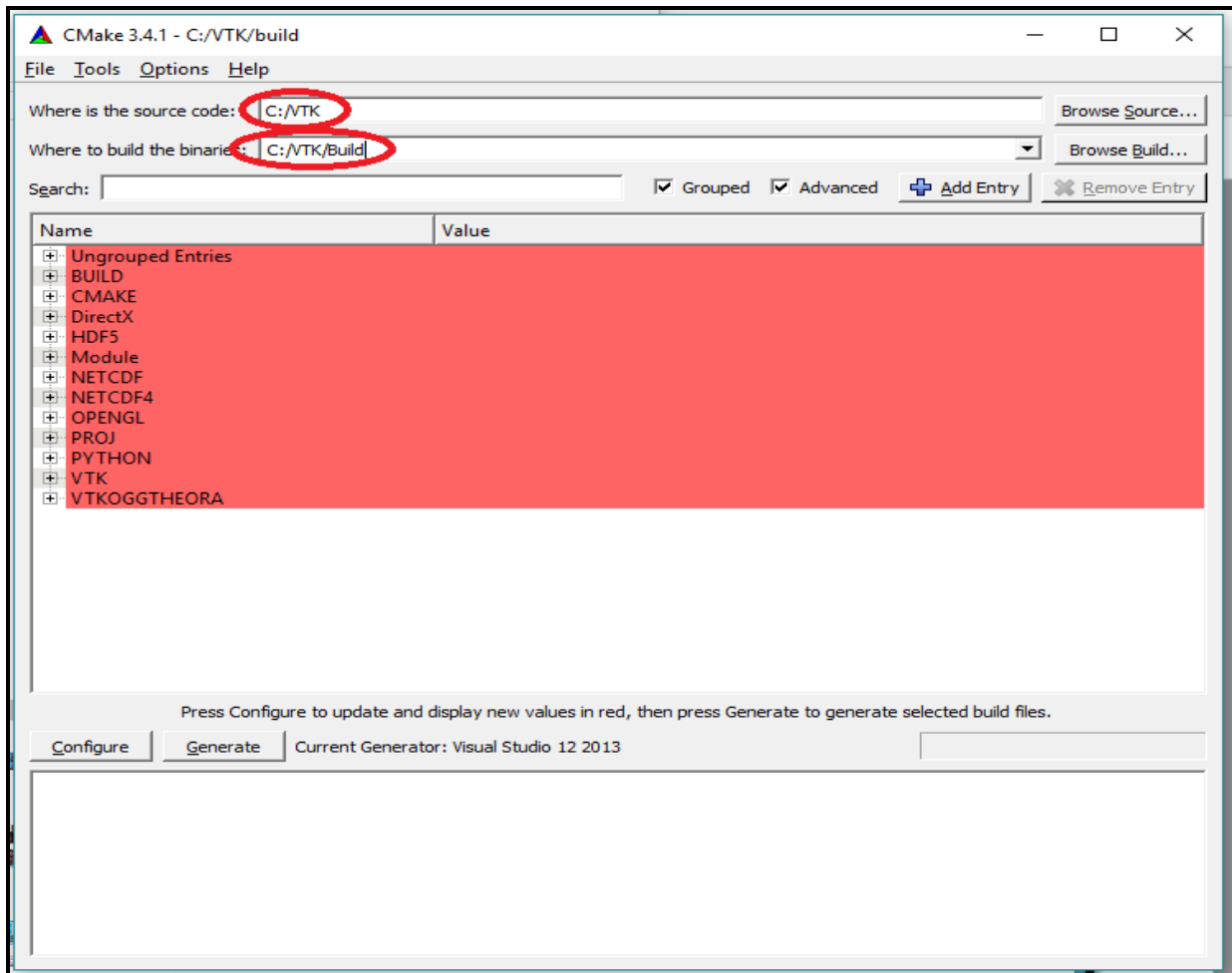


Figura 20: Construindo arquivos com CMake

As seguintes opções foram selecionadas:

- BUILD_EXAMPLES
- BUILD_SHARED_LIBS
- VTK_USE_GUISUPPORT
- VTK_USE_QT

Em seguida, o botão “*Configure*” deve ser pressionado novamente, logo após, o “*Generate*”.

Com a conclusão da geração dos arquivos, o resultado será um arquivo “VTK.sln”. Esse arquivo foi aberto com o Visual Studio e executado com o “*Build Solution*”, logo após, foi executado com o “*build*” no arquivo “INSTALL.vcxproj”.

Esse arquivo é o responsável de criar os arquivos de inclusão. Na Figura 19 é ilustrado o procedimento de geração dos arquivos por meio da IDE do Visual Studio:

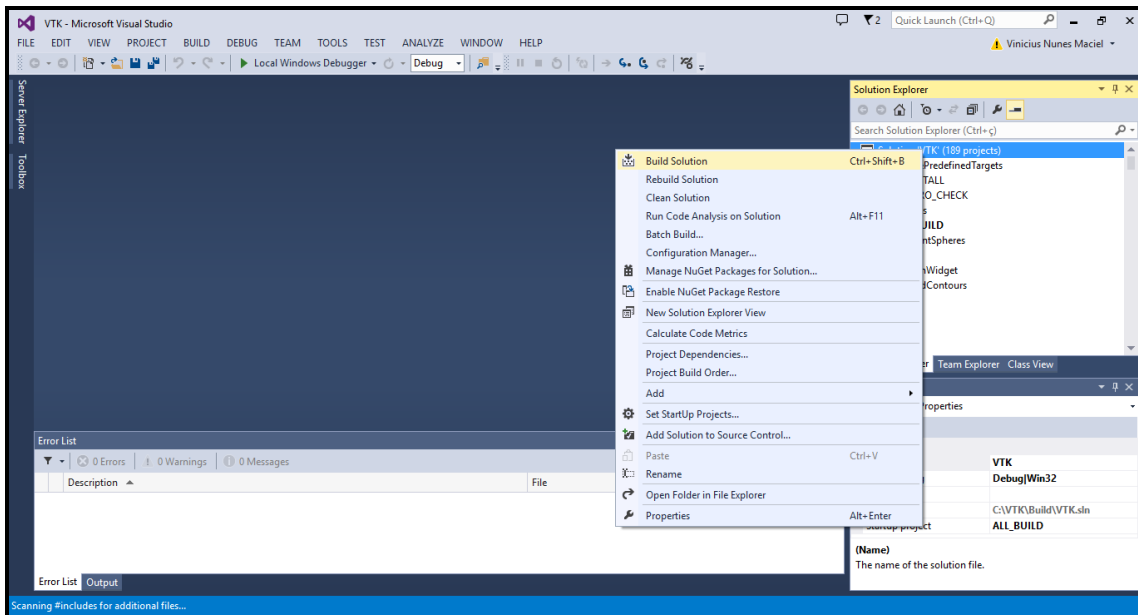


Figura 21: Construindo arquivos com Visual Studio

Dentro do diretório “C:\VTK\Build\bin\Debug”, foram gerados os arquivos: “QVTKWidgetPlugin.lib” e “QVTKWidgetPlugin.dll”. Os arquivos foram copiados para o diretório “C:\Qt\Qt5.4.1\5.4\msvc2013_opengl\plugins\designer”, permitindo que o *plugin* customizado seja utilizado como *Widget* do VTK no Qt.

Por fim, a última etapa da configuração do VTK deve-se modificar as variáveis de ambiente:

- VTK_DIR: “C:\VTK\Build”
- Adicionado “%VTK_DIR%\bin\Debug” e “C:\VTK\bin” na variável de ambiente PATH

A configuração do ITK é semelhante ao do VTK. Deve-se criar uma pasta “*build*” e a instrução CMake deve ser executada com a pasta de origem do ITK com o “CMakeLists.txt” e a pasta destino que é a pasta “*build*”.

A seguinte opção foi selecionada:

- BUILD_SHARED_LIBS

Em seguida, o botão “*Configure*” foi pressionado novamente, logo após, o “*Generate*”.

Após a geração dos arquivos, foi executado o arquivo “ITK.sln” no Visual Studio 2013 com o “*Build Solucion*” e executado o “*build*” no arquivo “INSTALL.vcxproj”.

Por fim, a última etapa da configuração do ITK deve-se modificar as variáveis de ambiente:

- ITK_DIR: “C:\ITK\Build”
- Adicionado “%ITK_DIR%\bin\Debug” e “C:\ITK\bin” na variável de ambiente PATH

Para ser possível usar o VTK e o ITK no Qt Creator foi necessário incluir o diretório onde estão os arquivos de inclusão (os arquivos .h) e as bibliotecas (os arquivos .lib) no arquivo .pro do projeto Qt. O código fonte completo pode ser consultado no Apêndice A.

Após a alteração do arquivo .pro, deve-se executar o “*qmake*” no projeto Qt para as configurações sejam aplicadas ao projeto selecionado.

4.2 IMPLEMENTAÇÃO

4.2.1 Método de visualizar imagem e séries de imagens DICOM

Por meio de uma variável do tipo `vtkDICOMImageReader` pode-se receber um arquivo com a extensão .dcm. E uma variável do tipo `vtkImageViewer2` é responsável em receber um arquivo de entrada, carregar na memória e colocar para visualização na *Widget* do VTK.

Elas são utilizadas tanto para uma imagem só, quanto para uma série de imagens. O que determina o que será inserido para visualização são os métodos `setFileName` e `setDirectoryName`. Se for passado um diretório de séries de imagens pelo método `setDirectoryName` todos os arquivos da série serão carregados, caso for um único arquivo pelo `setFileName` somente ele será carregado.

O trecho do código demonstra como é feita a leitura de um único arquivo por meio do VTK:

```
//...
//Variáveis Globais
//...

vtkSmartPointer<vtkDICOMImageReader> readerDCMSeries = new
vtkSmartPointer<vtkDICOMImageReader>::New();
vtkSmartPointer<vtkImageViewer2> imageViewDCMSeriesX = new
vtkSmartPointer<vtkImageViewer2>::New();
```

```

//...

void MainWindow::drawDICOMImg(std::string fileDICOM)
{
    single = true;
    series = false;

    readerDCMSeries->SetFileName(fileDICOM.c_str());
    std::cout << fileDICOM.c_str();
    readerDCMSeries->Update();
    imageViewDCMSeriesX->SetInputConnection(readerDCMSeries-
>GetOutputPort());
    imageViewDCMSeriesX->SetRenderWindow(ui->qvtkWidget-
>GetRenderWindow());
    imageViewDCMSeriesX->Render();
    imageViewDCMSeriesX->GetRenderer()->ResetCamera();
    imageViewDCMSeriesX->Render();

    if(single == true){
        ui->hSliderDCM->hide();
    }
    else ui->hSliderDCM->show();
    on_actionAxial_toggled(false);
    on_actionSagital_toggled(false);
    on_actionCoronal_toggled(false);
}

```

O trecho do código demonstra como é feita a leitura de séries de arquivos por meio do VTK:

```

//...
//Variáveis Globais
//...

vtkSmartPointer<vtkDICOMImageReader> readerDCMSeries = new
vtkSmartPointer<vtkDICOMImageReader>::New();
vtkSmartPointer<vtkImageViewer2> imageViewDCMSeriesX = new
vtkSmartPointer<vtkImageViewer2>::New();

//...

void MainWindow::drawDCMSeries(std::string folderDCM)
{
    single = false;
    series = true;

    readerDCMSeries->SetDirectoryName(folderDCM.c_str());
    readerDCMSeries->Update();
    imageViewDCMSeriesX->SetInputConnection(readerDCMSeries-
>GetOutputPort());
    imageViewDCMSeriesX->SetRenderWindow(ui->qvtkWidget-
>GetRenderWindow());
    mMinSliderX = imageViewDCMSeriesX->GetSliceMin();
    mMaxSliderX = imageViewDCMSeriesX->GetSliceMax();
    ui->hSliderDCM->setMinimum(mMinSliderX);
}

```

```

ui->hSliderDCM->setMaximum (mMaxSliderX) ;

imageViewDCMSeriesX->Render () ;
imageViewDCMSeriesX->GetRenderer ()->ResetCamera () ;
imageViewDCMSeriesX->Render () ;
if (series == true) {
    ui->hSliderDCM->show () ;
    ui->lcdNumber->show () ;
}
else {ui->hSliderDCM->hide () ;}
on_actionAxial_toggled (true) ;
on_actionSagital_toggled (false) ;
on_actionCoronal_toggled (false) ;
}

```

4.2.2 Método de leitura de imagem e séries de imagens DICOM e extração de tags

Na variável do tipo itk::Image é determinado o tipo de arquivo será usado, neste caso o tipo de pixel e a dimensão são apropriados para o tipo de arquivo .dcm. A variável do tipo itk::ImageSeriesReader é o responsável de carregar a imagem na memória. Para que se possa extrair suas informações foi preciso criar uma variável do tipo itk::GDCMImageIO para determinar o formato do arquivo e assim proporcionando métodos próprios para trabalho com o arquivo dicom.

O trecho do código a seguir demonstra como é feita a leitura de séries de arquivos e extrair suas tags através do ITK:

```

void MainWindow::InfoDCM (std::string folderDCM)
{
    ui->idPaciente->setText ("");
    ui->nomePaciente->setText ("");
    ui->bdPaciente->setText ("");
    ui->sexPaciente->setText ("");
    ui->agePaciente->setText ("");
    ui->dtStudy->setText ("");
    ui->modality->setText ("");
    ui->fabric->setText ("");
    ui->clinica->setText ("");
    ui->desc->setText ("");
    ui->medico->setText ("");
    ui->operator_2->setText ("");
    ui->slidew->setText ("");
    ui->slidel->setText ("");
    ui->slidef->setText ("");
}

```

```

typedef signed short PixelType;
const unsigned int Dimension = 3;

typedef itk::Image< PixelType, Dimension > ImageType;
typedef itk::ImageSeriesReader< ImageType > ReaderType;

ReaderType::Pointer reader = ReaderType::New();

typedef itk::GDCMImageIO      ImageIOType;

ImageIOType::Pointer dicomIO = ImageIOType::New();

reader->SetImageIO( dicomIO );

typedef itk::GDCMSeriesFileNames      NamesGeneratorType;

NamesGeneratorType::Pointer nameGenerator =
NamesGeneratorType::New();

nameGenerator->SetInputDirectory(folderDCM.c_str());

typedef std::vector<std::string>      FileNamesContainer;
FileNamesContainer fileNames = nameGenerator-
>GetInputFileNames();

reader->SetFileNames( fileNames );

try
{
    // Software Guide : BeginCodeSnippet
    reader->Update();
    // Software Guide : EndCodeSnippet
}
catch (itk::ExceptionObject &ex)
{
    std::cout << ex << std::endl;
}

typedef itk::MetaDataDictionary      DictionaryType;

const DictionaryType & dictionary = dicomIO-
>GetMetaDataDictionary();

typedef itk::MetaDataObject< std::string > MetaDataStringType;

DictionaryType::ConstIterator itr = dictionary.Begin();
DictionaryType::ConstIterator end = dictionary.End();

while( itr != end )
{
    itk::MetaDataObjectBase::Pointer entry = itr->second;

    MetaDataStringType::Pointer entryvalue =
        dynamic_cast<MetaDataStringType *>( entry.GetPointer() );

    if( entryvalue )

```

```

{
    std::string tagkey    = itr->first;
    std::string tagvalue = entryvalue->GetMetaDataObjectValue();
    std::cout << tagkey << " = " << tagvalue << std::endl;
    if( tagkey == "0010|0020" )
    {
        ui->idPaciente->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0010|0010" )
    {
        ui->nomePaciente-
>setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0010|0030" )
    {
        ui->bdPaciente->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0010|0040" )
    {
        ui->sexPaciente-
>setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0010|1010" )
    {
        ui->agePaciente-
>setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|0020" )
    {
        ui->dtStudy->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|0060" )
    {
        ui->modality->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|0070" )
    {
        ui->fabric->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|0080" )
    {
        ui->clinica->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|1030" )
    {
        ui->desc->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|1060" )
    {
        ui->medico->setText (QString::fromStdString(tagvalue));
    }
    if( tagkey == "0008|1070" )
    {
        ui->operator_2->setText (QString::fromStdString(tagvalue));
    }
}

++itr;
}
}

```

4.3 INTERFACES DA APLICAÇÃO

Na interface principal do visualizador, contem um menubar com as opções:

- Arquivo - Abrir Arquivo DICOM, Abrir Series DICOM, Exportar, Gerar Laudo e Sair;
- Ferramentas - Contraste, Zoom, Rotação, Escrever, Visão Axial, Visão Sagital, Visão Coronal e Exibir Tags;
- Sobre.

Além de uma widget visualização, Slider para seleção de fatias (É exibido somente quando há uma série de arquivos abertos), display LCD para a exibição das fatias, uma tabela de informações do paciente e uma outra com informações do exame.

A Figura 22 ilustra a interface principal do visualizador:

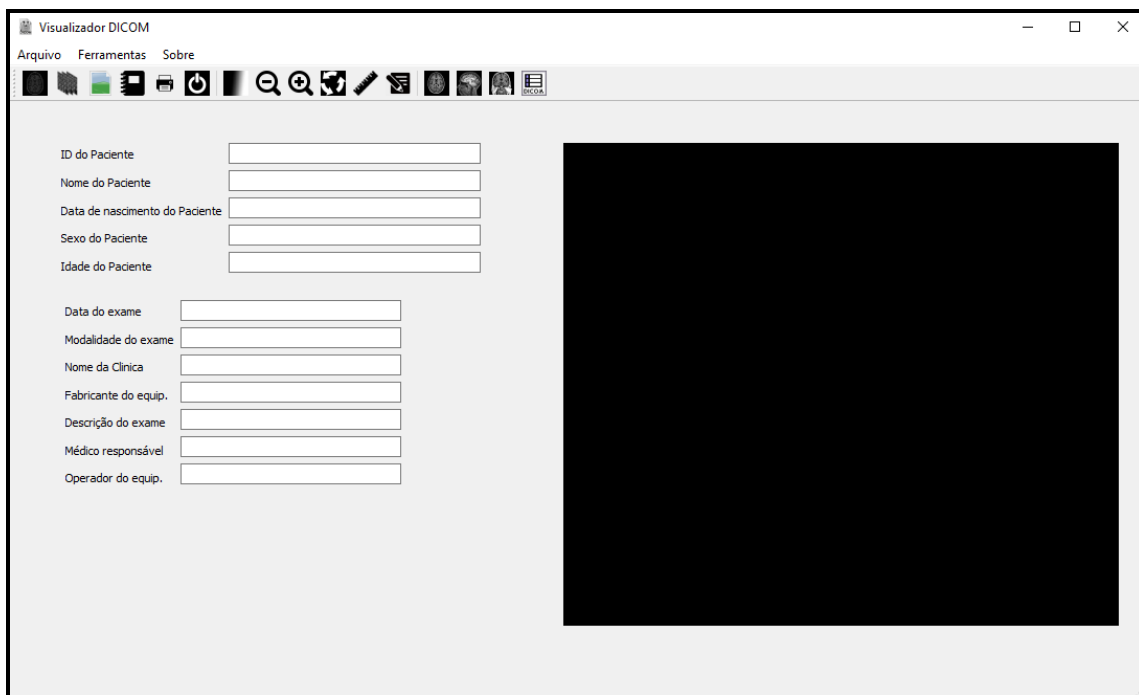


Figura 22: Interface da Tela Principal

A Figura 23 ilustra uma interface para a visualização de uma única imagem DICOM e suas informações.

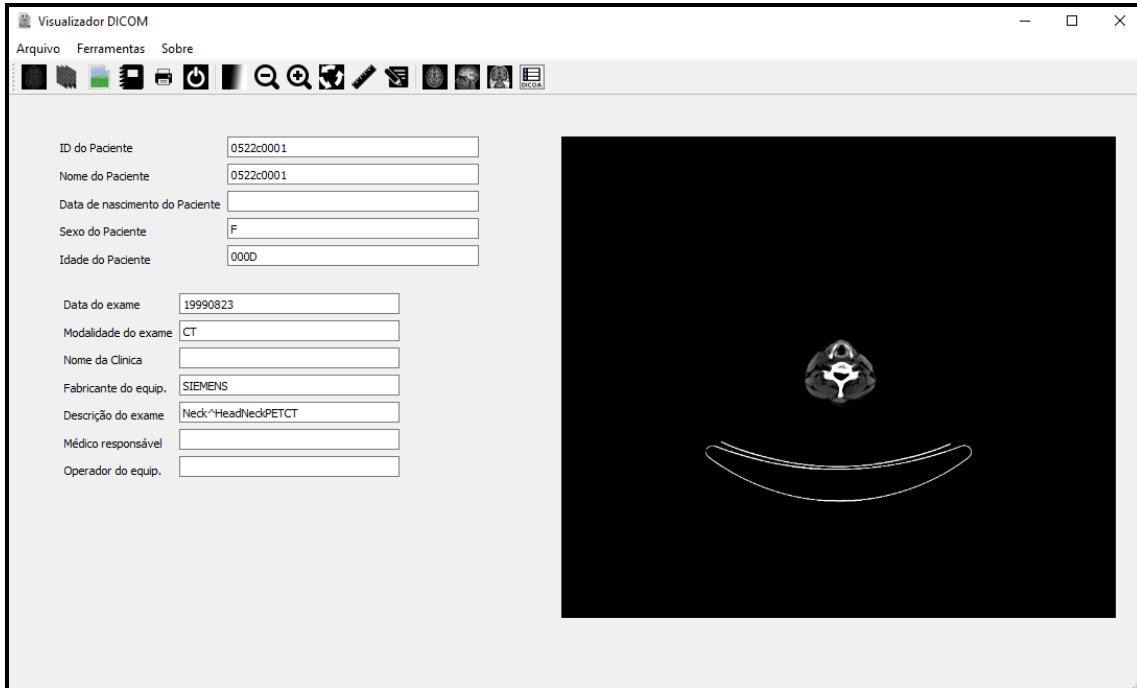


Figura 23: Visualização de imagem DICOM e suas informações

A Figura 24 ilustra a interface para a visualização de séries de arquivos DICOM.

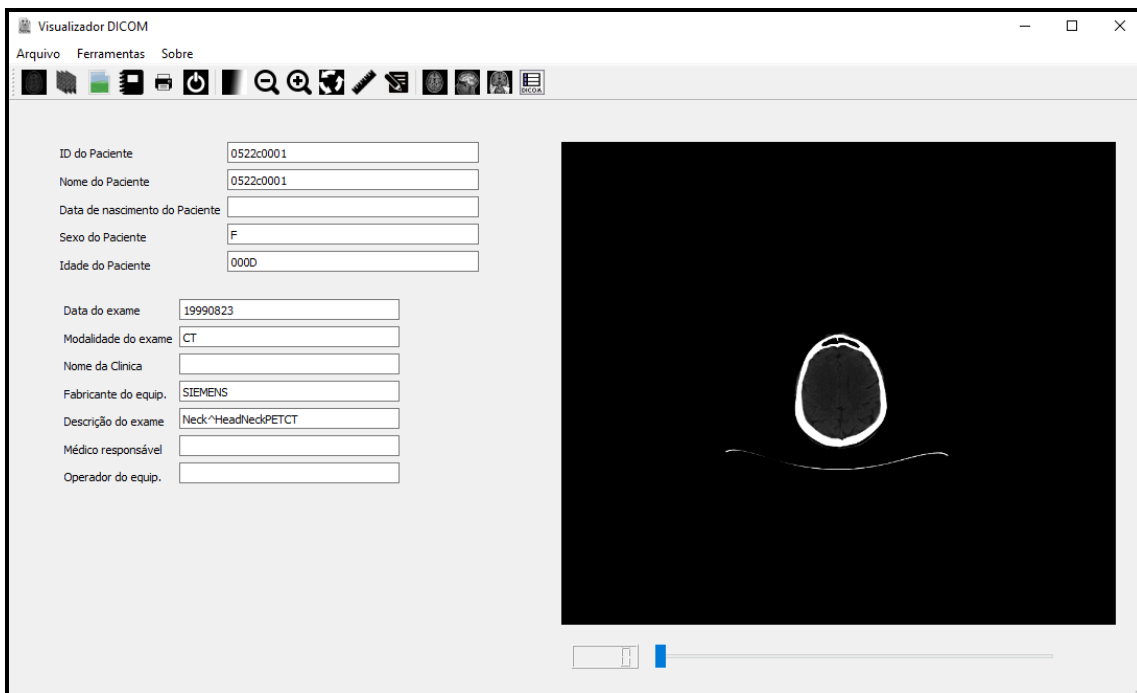


Figura 24: Visualização de série de imagens DICOM e suas informações

Todo arquivo DICOM tem seu cabeçalho contendo algumas informações sobre o exame, que são as *Tags*. A Figura 25 ilustra a interface exibindo algumas dessas informações contidas no cabeçalho do arquivo.

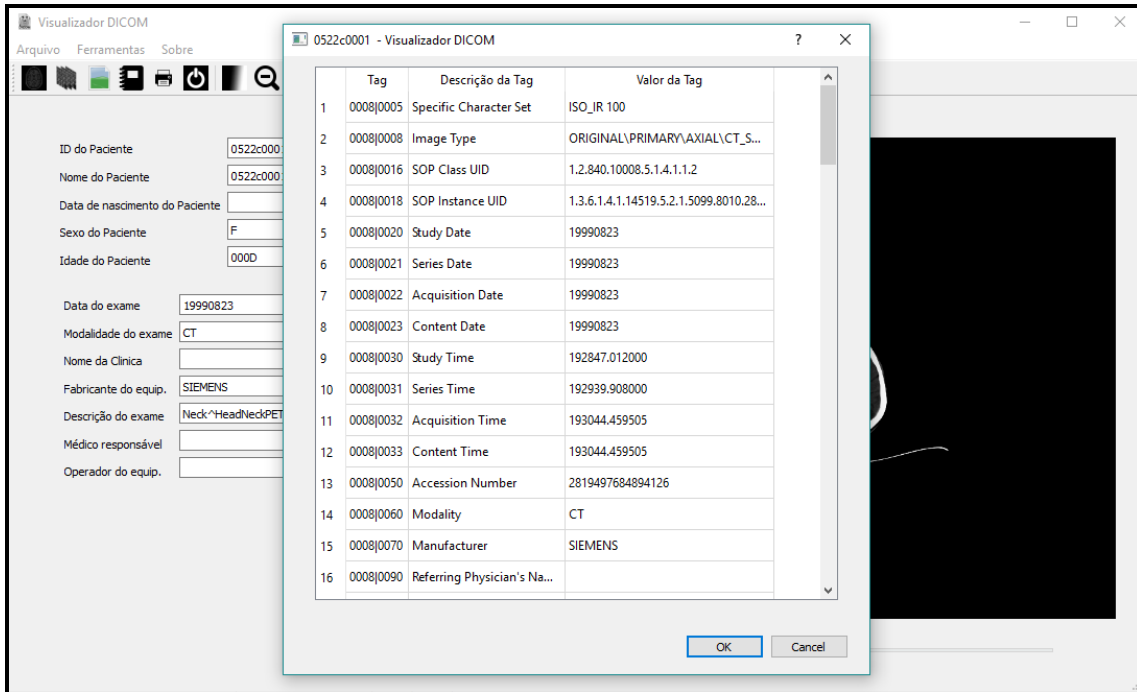


Figura 25: Exibição das tags do arquivo

Por padrão, as imagens DICOM de exames como CT e MR são geradas na orientação Axial, porém, utilizando o “VTK” é possível gerar outras orientações de visualização das imagens, mas apenas com essas duas modalidades de exames. Nas interfaces a seguir será mostra a visualização na orientação Sagital e Coronal.

A Figura 26 ilustra a interface para a visualização na orientação Sagital.

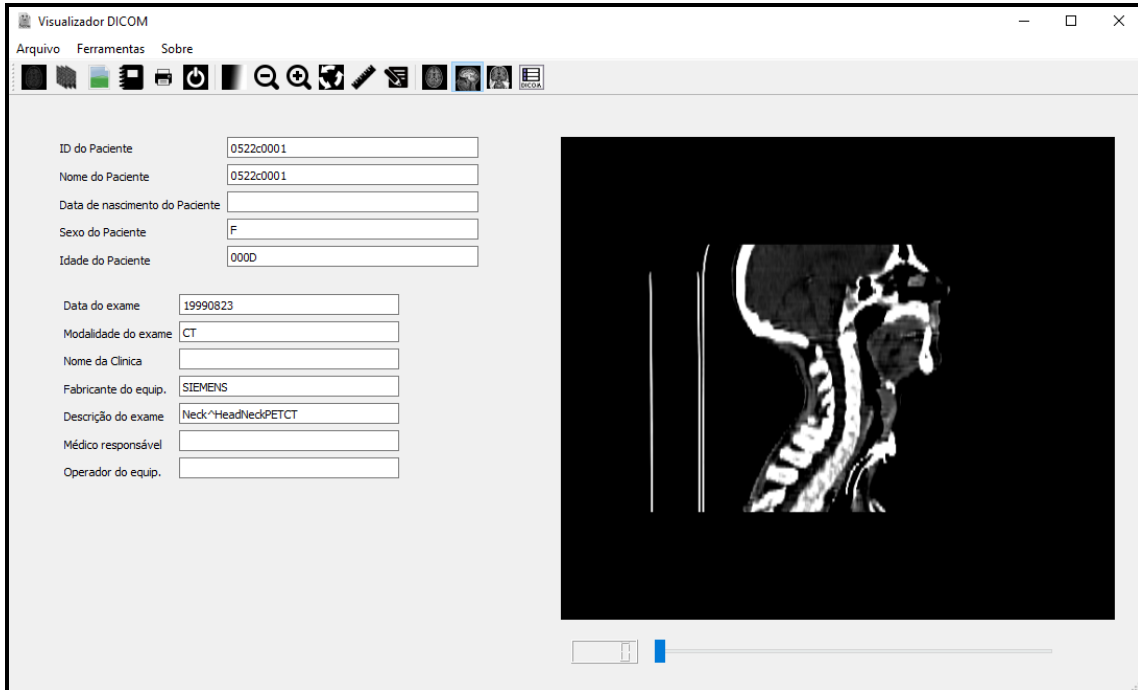


Figura 26: Visualização na orientação Sagital

A Figura 27 ilustra a interface para a visualização na orientação Coronal.

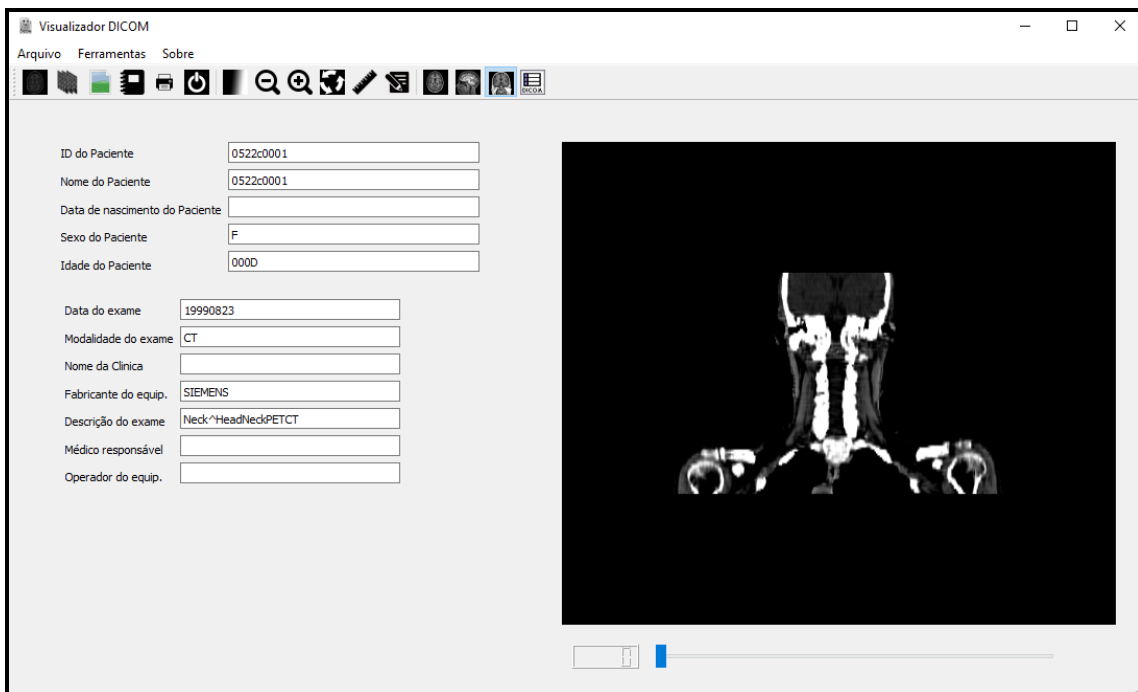


Figura 27: Visualização na orientação Coronal

A seguir serão mostradas visualizações com modificações:

A Figura 28 ilustrada a interface para a visualização com contraste aplicado.

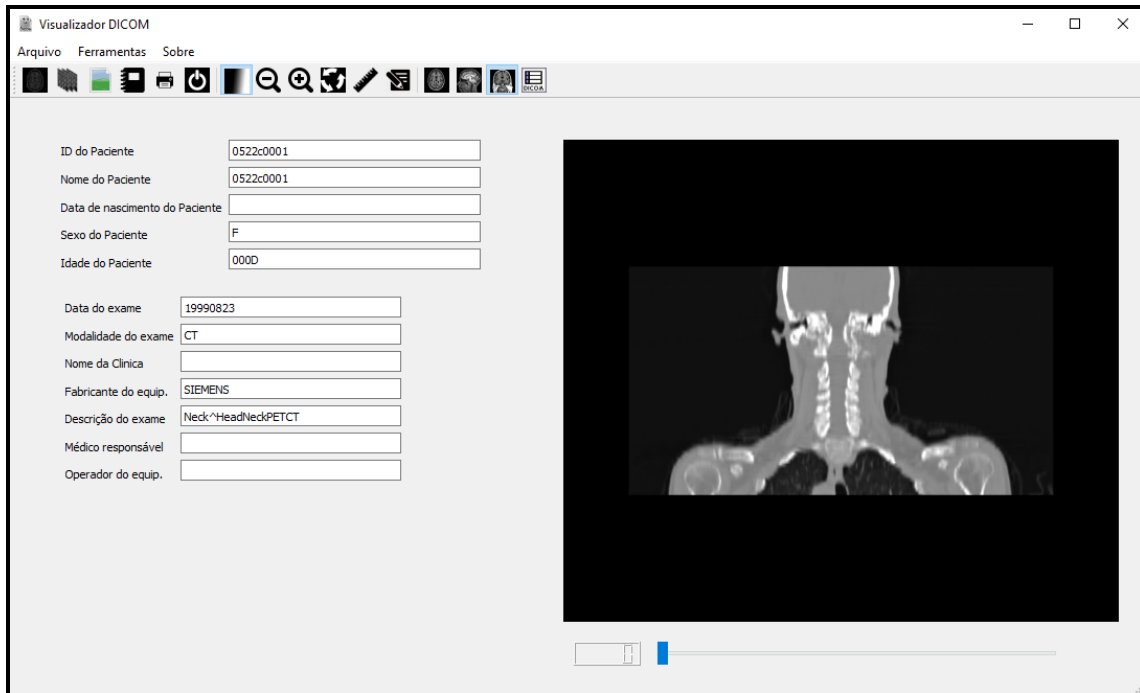


Figura 28: Visualização com contraste aplicado

A Figura 29 ilustra a interface para a visualização com zoom aplicado.

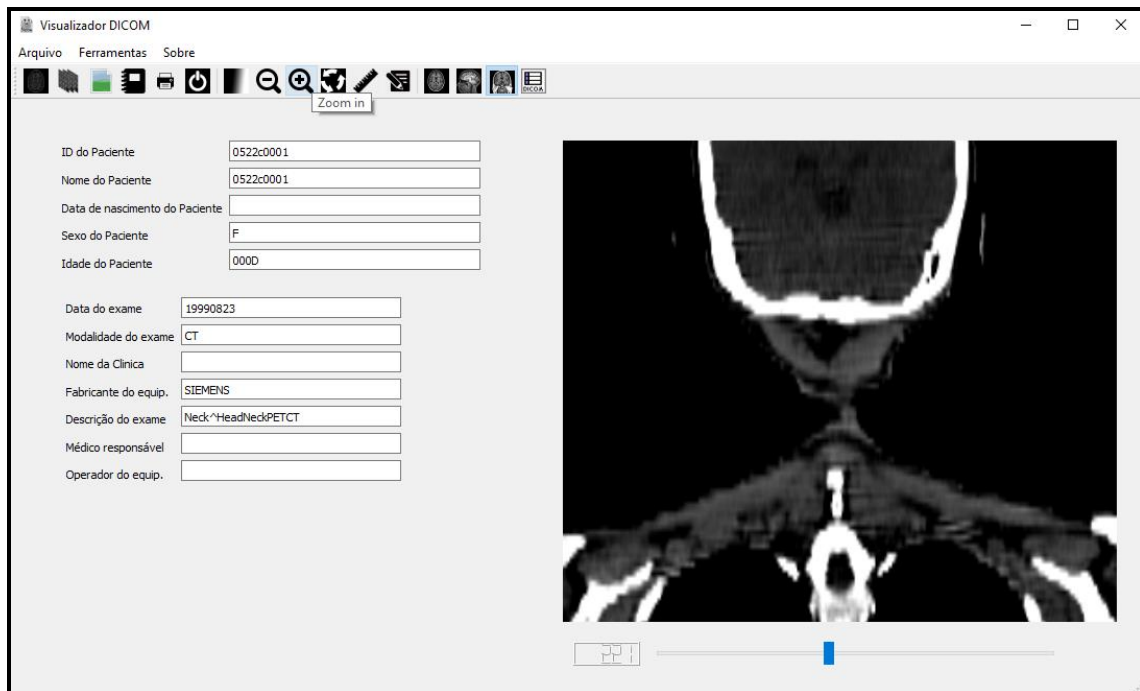


Figura 29: Visualização com zoom aplicado

A Figura 30 ilustra a interface do editor de laudos.

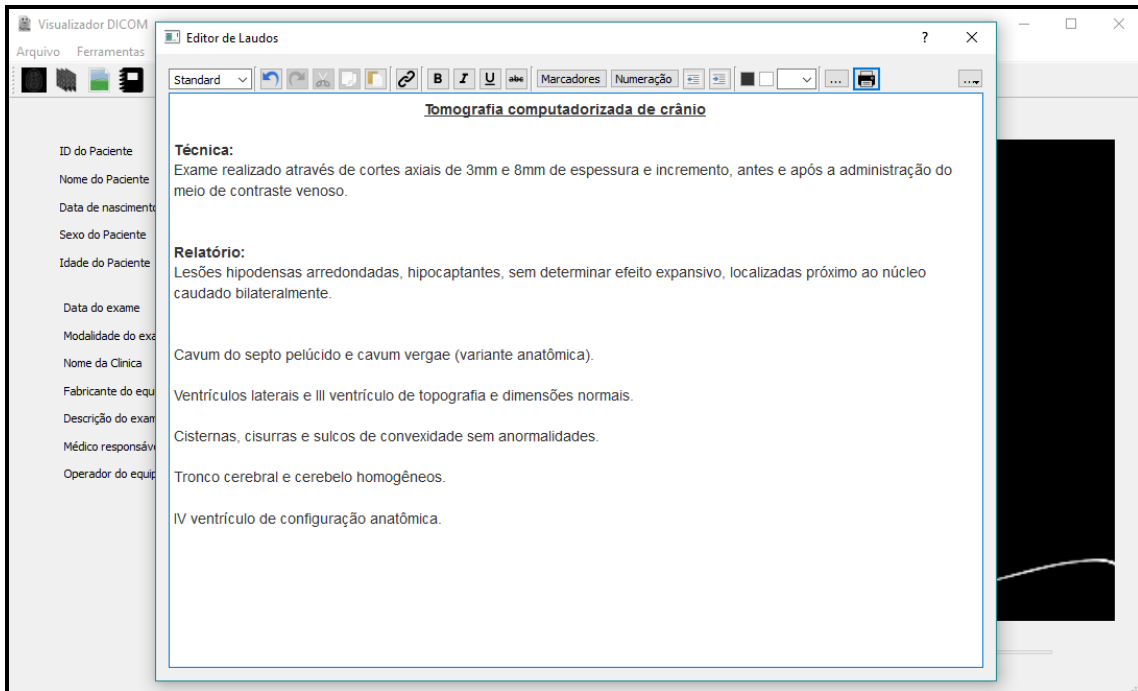


Figura 30: Tela de Laudos

Foram acrescentadas mais duas funcionalidades ao final do trabalho, a aplicação de rotação e medida.

A Figura 31 ilustra a interface para a visualização com rotação aplicada.

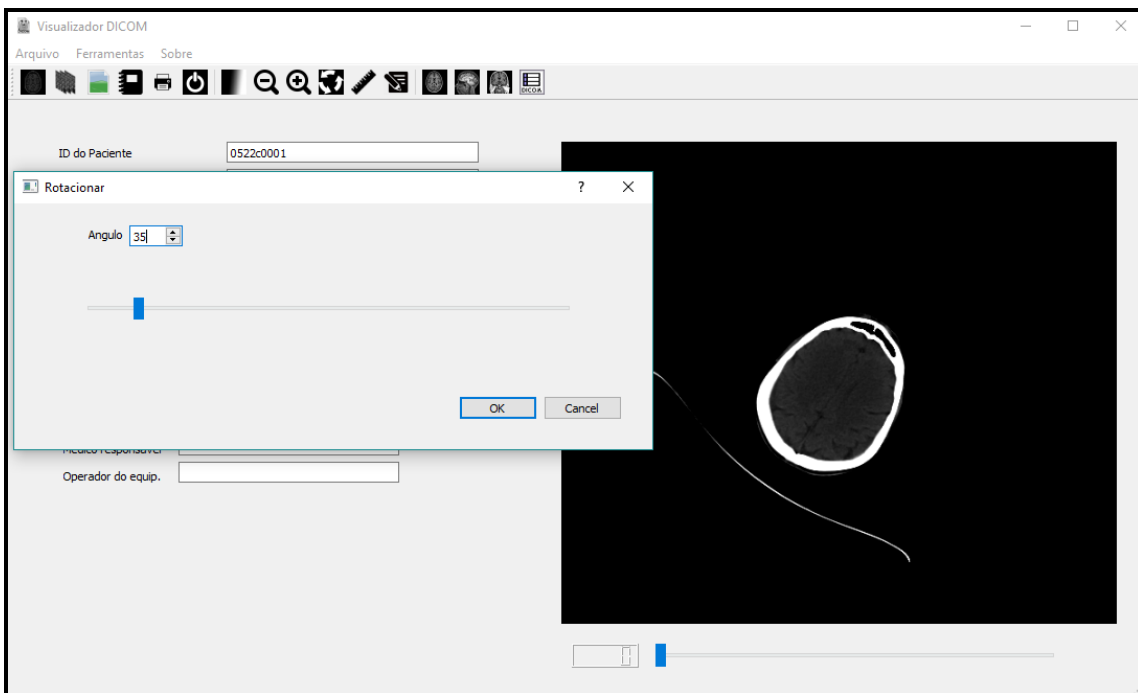


Figura 31: Visualização com rotação aplicada

A Figura 32 ilustra a interface para a visualização com medida aplicada.

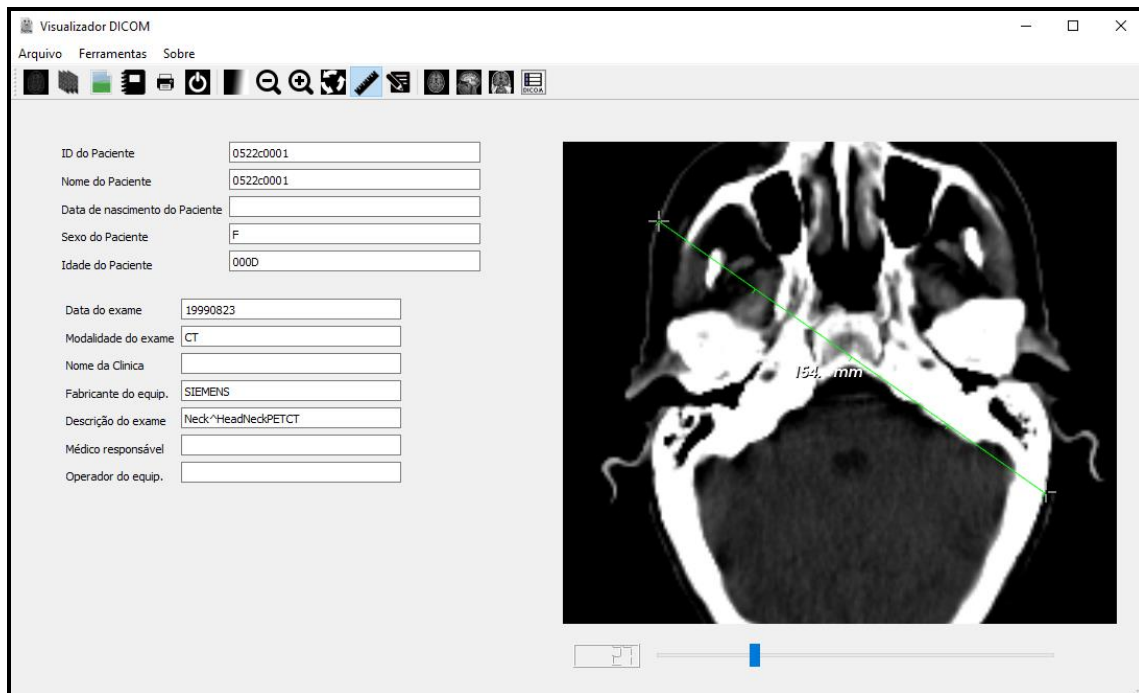


Figura 32: Visualização com medida aplicada

5 – CONCLUSÃO

A utilização de tecnologia na medicina tem ajudado muito no seu avanço. Além disso, como a medicina não é uma ciência exata, as tecnologias podem auxiliar na resolução de problemas e no melhor resultado dos diagnósticos de pacientes. Com isso, evita-se que erros grotescos causem problemas graves ao paciente e/ou a seu diagnóstico.

A imagiologia médica utiliza equipamentos tecnológicos para obter uma imagem com a maior quantidade possível por meio de vários pré-processamentos computacionais. Os equipamentos geram imagens em um formato digital que podem ser facilmente processadas por meios computacionais.

Muitas clínicas ainda não acompanham esses avanços e recursos tecnológicos, ainda utilizam métodos antigos, como imprimir as imagens em chapas para análise.

Por meio de um programa que manipule exames radiológicos de maneira digital, uma melhor análise pode ser conduzida e resultados precisos podem ser obtidos. Outro exemplo é a capacidade de enviar o exame via Internet para que outro radiologista possa analisá-la, bem como distintas vantagens de se trabalhar descentralizado e na Web.

Por fim, pode-se perceber que ainda há poucos softwares dedicados no mercado com essas finalidades, devido a esta carência, muitas empresas são obrigadas a buscar soluções importadas que apoiam o diagnóstico e demais atividades relacionadas à análise de imagens médicas.

5.1 – TRABALHOS FUTUROS

A partir deste trabalho é possível dar continuidade, desenvolvendo um sistema completo que utilize mais técnicas para processamentos de imagens para uma melhor qualidade dos arquivos DICOM, visualização volumétrica e 3D, comunicação com rede, armazenamento e comunicação cliente-servidor.

REFERÊNCIAS

AZEVEDO-MARQUES, Paulo Mazzoncini de. **Diagnóstico auxiliado por computador na radiologia**. In: Radiol Bras, São Paulo, v. 34, n. 5, p. 285-293, Oct. 2001. Disponível em: < http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-39842001000500008&lng=en&nrm=iso > Acesso em 12/02/2016.

CMake: **Build, Test and Package Your Software With CMake**. Disponível em: <<https://cmake.org/>> Acesso em 26/07/2016.

DOI, Kunio. **Computer-aided diagnosis and its potential impact on diagnostic radiology**. In: Doi K, MacMahon H, Giger ML, Hoffmann KR, eds. Computer-aided diagnosis in medical imaging. Amsterdam: Elsevier Science, 1999, p.11 - 20.

FARIA, Diogo. **Análise e Processamento de Imagem**. Trabalho Prático, Porto: Faculdade de Engenharia da Universidade do Porto, 2010. 44p.

FENELON, Sandro. **Aspectos ético-legais em Imaginologia**. Radiol Bras, São Paulo, v. 36, n.1, p.03-06, Jan. 2003. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-39842003000100001&lng=en&nrm=iso> Acesso em 10/03/2016.

GONZALEZ, R. C.; WOODS, R. E.. **Digital Image Processing**. Pearson Education, Inc., 2008.

ITK: **Insight Segmentation and Registration Toolkit**. Disponível em: < <http://www.itk.org/itkindex.html> > Acesso em 20/12/2015.

MANSSOUR, I. H.; FREITAS, C. M. D. S., Visualização Volumétrica, **RITA - Revista de Informática Teórica e Aplicada**, V. IX, n. 2, 2002, p.97-126.

MONTEIRO, Denyse Nascimento Barcellos Monteiro. **Estudo sobre a Visualização de Imagens Médicas obtida por Exames Virtuais**. Dissertação de Mestrado, Niterói: Universidade Federal Fluminense, 2005. 124p.

MSDN: Microsoft Developer Network. **Guia de Introdução ao Visual Studio**. Disponível em: < [https://msdn.microsoft.com/pt-br/library/ms165079\(v=vs.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms165079(v=vs.120).aspx) > Acesso em 18/02/2016.

NCBI: **National Center for Biotechnology Information: Managing DICOM images: Tips and tricks for the radiologist**. Disponível em: < <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3354356/> > Acesso em 06/03/2016.

NEMA: **NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION: DICOM Homepage**. Disponível em: <<http://dicom.nema.org>> Acesso em 08/01/2016

OLIVEIRA, Lutero Marques de. **Radiologia e Diagnóstico por Imagem - Ética, Normas, Direitos e Deveres dos Médicos Imaginologistas: Código de Processo Ético Profissional em Radiologia e Diagnóstico por Imagem: Colégio Brasileiro de Radiologia e Diagnóstico por Imagem (CBR)**, 2012. 136 p.

PORTO, Fuad Nacif. **Análise de sensibilidade de um sistema CAD para Mamografia Digital**. Dissertação de Mestrado, Belo Horizonte: Universidade Federal de Minas Gerais, 2010. 69p.

RIBEIRO, Brenno Bernardes. **Introdução ao Protocolo DICOM e Implementação de um Visualizador Compatível**. Trabalho de Conclusão de Curso, Pará: Universidade Federal do Pará, 2013. 64p.

SAITO, Priscila Tiemi Maeda. **Otimização do Processamento de Imagens Médicas utilizando a Computação Paralela**. Trabalho de Conclusão de Curso, Marília: Centro Universitário Eurípides de Marília – UNIVEM, 2007. 160p.

SANCHES, Ionildo José. **Sobreposição de imagens de termografia e ressonância magnética: uma nova modalidade de imagem médica tridimensional**. Tese de Doutorado, Curitiba: Universidade Tecnológica Federal do Paraná - UTFPR, 2009. 168p.

SEIXAS, Flávio Luiz. **Diagnóstico Auxiliado por Computador**. Trabalho de Conclusão de Curso, Niterói: Universidade Federal Fluminense, 2005. 40p.

SILVA, Diego de Souza. **Processamento de Imagem Técnicas e Aplicações**. Disponível em < <http://pt.slideshare.net/diesousil/processamento-de-imagem> > Acesso em 27/07/2016.

STROUSTRUP, Bjarne. **The C++ Programming Language**. Disponível em <<http://www.stroustrup.com/C++.html> > Acesso em 04/04/2016.

THELIN, Johan. **Foundations of Qt Development**, 1ed. Berkeley. Editora APRESS, 2007. 513p.

VTK: **Visualization Toolkit**. Disponível em: < <http://www.vtk.org/> > Acesso em 02/10/2015

APÊNDICE A

Código fonte para configuração do VTK e ITK no Qt Creator.

```

#-----
#
# Project created by QtCreator 2016-07-20T16:13:10
#
#-----

QT      += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = Testell
TEMPLATE = app

SOURCES += main.cpp\
           mainwindow.cpp

HEADERS  += mainwindow.h

FORMS    += mainwindow.ui

INCLUDEPATH += "C:\\VTK\\include\\vtk-6.2\\"

LIBS += -LC:\\VTK\\Build\\bin\\Debug
LIBS += -LC:\\VTK\\lib

LIBS += -lvtkCommonCore-6.2
LIBS += -lvtkCommonExecutionModel-6.2
LIBS += -lvtksys-6.2
LIBS += -lQVTKWidgetPlugin           # QT RELATED
LIBS += -lvtkChartsCore-6.2
LIBS += -lvtkViewsQt-6.2             # QT Related
LIBS += -lvtkInteractionWidgets-6.2
LIBS += -lvtkInfovisCore-6.2
LIBS += -lvtkRenderingCore-6.2
LIBS += -lvtkRenderingOpenGL-6.2
LIBS += -lvtkImagingCore-6.2
LIBS += -lvtklibxml2-6.2
LIBS += -lvtkDICOMParser-6.2
LIBS += -lvtkpng-6.2
LIBS += -lvktiff-6.2
LIBS += -lvtkzlib-6.2
LIBS += -lvtkjpeg-6.2
LIBS += -lvtkalglib-6.2
LIBS += -lvtkexpat-6.2
LIBS += -lvtkverdict-6.2
LIBS += -lvtkmetaio-6.2
LIBS += -lvtkNetCDF-6.2
LIBS += -lvtksqlite-6.2
LIBS += -lvtkexoIIC-6.2
LIBS += -lvtkftgl-6.2

```

```

LIBS += -lvtkfreetype-6.2
LIBS += -lvtkFiltersSources-6.2
LIBS += -lvtkGUISupportQt-6.2          #QT related

LIBS += -lvtkalglib-6.2
LIBS += -lvtkChartsCore-6.2
LIBS += -lvtkCommonColor-6.2
LIBS += -lvtkCommonComputationalGeometry-6.2
LIBS += -lvtkCommonCore-6.2
LIBS += -lvtkCommonDataModel-6.2
LIBS += -lvtkCommonExecutionModel-6.2
LIBS += -lvtkCommonMath-6.2
LIBS += -lvtkCommonMisc-6.2
LIBS += -lvtkCommonSystem-6.2
LIBS += -lvtkCommonTransforms-6.2
LIBS += -lvtkDICOMParser-6.2
LIBS += -lvtkDomainsChemistry-6.2
LIBS += -lvtkexoIIc-6.2
LIBS += -lvtkexpat-6.2
LIBS += -lvtkFiltersAMR-6.2
LIBS += -lvtkFiltersCore-6.2
LIBS += -lvtkFiltersExtraction-6.2
LIBS += -lvtkFiltersFlowPaths-6.2
LIBS += -lvtkFiltersGeneral-6.2
LIBS += -lvtkFiltersGeneric-6.2
LIBS += -lvtkFiltersGeometry-6.2
LIBS += -lvtkFiltersHybrid-6.2
LIBS += -lvtkFiltersHyperTree-6.2
LIBS += -lvtkFiltersImaging-6.2
LIBS += -lvtkFiltersModeling-6.2
LIBS += -lvtkFiltersParallel-6.2
LIBS += -lvtkFiltersParallelImaging-6.2
LIBS += -lvtkFiltersProgrammable-6.2
LIBS += -lvtkFiltersSelection-6.2
LIBS += -lvtkFiltersSMP-6.2
LIBS += -lvtkFiltersSources-6.2
LIBS += -lvtkFiltersStatistics-6.2
LIBS += -lvtkFiltersTexture-6.2
LIBS += -lvtkFiltersVerdict-6.2
LIBS += -lvtkfreetype-6.2
LIBS += -lvtkftgl-6.2
LIBS += -lvtkGeovisCore-6.2
LIBS += -lvtkgl2ps-6.2
LIBS += -lvtkGUISupportQt-6.2
LIBS += -lvtkGUISupportQtOpenGL-6.2    #qt related
LIBS += -lvtkGUISupportQtSQL-6.2      #qt related
LIBS += -lvtkGUISupportQtWebKit-6.2   #qt related
LIBS += -lvtkhdf5_hl-6.2
LIBS += -lvtkhdf5-6.2
LIBS += -lvtkImagingColor-6.2
LIBS += -lvtkImagingCore-6.2
LIBS += -lvtkImagingFourier-6.2
LIBS += -lvtkImagingGeneral-6.2
LIBS += -lvtkImagingHybrid-6.2
LIBS += -lvtkImagingMath-6.2
LIBS += -lvtkImagingMorphological-6.2
LIBS += -lvtkImagingSources-6.2
LIBS += -lvtkImagingStatistics-6.2
LIBS += -lvtkImagingStencil-6.2
LIBS += -lvtkInfovisCore-6.2
LIBS += -lvtkInfovisLayout-6.2

```

```

LIBS += -lvtkInteractionImage-6.2
LIBS += -lvtkInteractionStyle-6.2
LIBS += -lvtkInteractionWidgets-6.2
LIBS += -lvtkIOAMR-6.2
LIBS += -lvtkIOCore-6.2
LIBS += -lvtkIOEnSight-6.2
LIBS += -lvtkIOExodus-6.2
LIBS += -lvtkIOExport-6.2
LIBS += -lvtkIOGeometry-6.2
LIBS += -lvtkIOImage-6.2
LIBS += -lvtkIOImport-6.2
LIBS += -lvtkIOInfovis-6.2
LIBS += -lvtkIOLegacy-6.2
LIBS += -lvtkIOLSDyna-6.2
LIBS += -lvtkIOMINC-6.2
LIBS += -lvtkIOMovie-6.2
LIBS += -lvtkIONetCDF-6.2
LIBS += -lvtkIOParallel-6.2
LIBS += -lvtkIOPLY-6.2
LIBS += -lvtkIOSQL-6.2
LIBS += -lvtkIOVideo-6.2
LIBS += -lvtkIOXML-6.2
LIBS += -lvtkIOXMLParser-6.2
LIBS += -lvtkjjpeg-6.2
LIBS += -lvtkjsoncpp-6.2
LIBS += -lvtklibxml2-6.2
LIBS += -lvtkLocalExample-6.2
LIBS += -lvtkmetaio-6.2
LIBS += -lvtkNetCDF_cxx-6.2
LIBS += -lvtkNetCDF-6.2
LIBS += -lvtkoggtheora-6.2
LIBS += -lvtkParallelCore-6.2
LIBS += -lvtkpng-6.2
LIBS += -lvtkproj4-6.2
LIBS += -lvtkRenderingAnnotation-6.2
LIBS += -lvtkRenderingContext2D-6.2
LIBS += -lvtkRenderingCore-6.2
LIBS += -lvtkRenderingFreeType-6.2
LIBS += -lvtkRenderingFreeTypeOpenGL-6.2
LIBS += -lvtkRenderingGL2PS-6.2
LIBS += -lvtkRenderingImage-6.2
LIBS += -lvtkRenderingLabel-6.2
LIBS += -lvtkRenderingLIC-6.2
LIBS += -lvtkRenderingLOD-6.2
LIBS += -lvtkRenderingOpenGL-6.2
LIBS += -lvtkRenderingQt-6.2
LIBS += -lvtkRenderingVolume-6.2
# LIBS += -lvtkRenderingVolumeAMR-6.2
LIBS += -lvtkRenderingVolumeOpenGL-6.2
LIBS += -lvtksqlite-6.2
LIBS += -lvtksys-6.2
LIBS += -lvtkTestingRendering-6.2
LIBS += -lvtktiff-6.2
LIBS += -lvtkverdict-6.2
LIBS += -lvtkViewsContext2D-6.2
LIBS += -lvtkViewsCore-6.2
# LIBS += -lvtkViewsGeovis-6.2
LIBS += -lvtkViewsInfovis-6.2
LIBS += -lvtkViewsQt-6.2
LIBS += -lvtkzlib-6.2

```

```
#qt related
```

```
#QT Related
```

```

INCLUDEPATH += "C:\\ITK\\include\\ITK-4.7\\"

LIBS += -LC:\\ITK\\Build\\bin\\Debug
LIBS += -LC:\\ITK\\lib

LIBS += -lITKBiasCorrection-4.7
LIBS += -lITKBioCell-4.7
LIBS += -lITKCommon-4.7
LIBS += -lITKDICOMParser-4.7
LIBS += -litkdouble-conversion-4.7
LIBS += -lITKEXPAT-4.7
LIBS += -lITKFEM-4.7
LIBS += -litkgdcmCommon-4.7
LIBS += -litkgdcmDICT-4.7
LIBS += -litkgdcmDSED-4.7
LIBS += -litkgdcmIOD-4.7
LIBS += -litkgdcmjpeg8-4.7
LIBS += -litkgdcmjpeg12-4.7
LIBS += -litkgdcmjpeg16-4.7
LIBS += -litkgdcmMSFF-4.7
LIBS += -lITKgiftiio-4.7
LIBS += -litkhdf5_cpp-4.7
LIBS += -litkhdf5-4.7
LIBS += -lITKIOBioRad-4.7
LIBS += -lITKIOBMP-4.7
LIBS += -lITKIOCSV-4.7
LIBS += -lITKIOGDCM-4.7
LIBS += -lITKIOGE-4.7
LIBS += -lITKIOGIPL-4.7
LIBS += -lITKIOHDF5-4.7
LIBS += -lITKIOImageBase-4.7
LIBS += -lITKIOIPL-4.7
LIBS += -lITKIOJPEG-4.7
LIBS += -lITKIOLSM-4.7
LIBS += -lITKIOMesh-4.7
LIBS += -lITKIOMeta-4.7
LIBS += -lITKIOMRC-4.7
LIBS += -lITKIONIFTI-4.7
LIBS += -lITKIONRRD-4.7
LIBS += -lITKIOPNG-4.7
LIBS += -lITKIOSiemens-4.7
LIBS += -lITKIOSpatialObjects-4.7
LIBS += -lITKIOStimulate-4.7
LIBS += -lITKIOTIFF-4.7
LIBS += -lITKIOTransformBase-4.7
LIBS += -lITKIOTransformHDF5-4.7
LIBS += -lITKIOTransformInsightLegacy-4.7
LIBS += -lITKIOTransformMatlab-4.7
LIBS += -lITKIOVTK-4.7
LIBS += -lITKIOXML-4.7
LIBS += -litkjpeg-4.7
LIBS += -lITKKLMMRegionGrowing-4.7
LIBS += -lITkLabelMap-4.7
LIBS += -lITKMesh-4.7
LIBS += -lITKMetaIO-4.7
LIBS += -litkNetlibSlatec-4.7
LIBS += -lITKniftiio-4.7
LIBS += -lITKNrrdIO-4.7
LIBS += -litkopenjpeg-4.7

```

```
LIBS += -lITKOptimizers-4.7
LIBS += -lITKOptimizersv4-4.7
LIBS += -lITKPath-4.7
LIBS += -litkpng-4.7
LIBS += -lITKPolynomials-4.7
LIBS += -lITKQuadEdgeMesh-4.7
LIBS += -lITKSpatialObjects-4.7
LIBS += -lITKStatistics-4.7
LIBS += -litksys-4.7
LIBS += -litktiff-4.7
LIBS += -litkv3p_lsqr-4.7
LIBS += -litkv3p_netlib-4.7
LIBS += -litkvcl-4.7
LIBS += -litkVideoCore-4.7
LIBS += -litkVideoIO-4.7
LIBS += -litkvnl_algo-4.7
LIBS += -litkvnl-4.7
LIBS += -lITKVNLInstantiation-4.7
LIBS += -lITKVTK-4.7
LIBS += -lITKVtkGlue-4.7
LIBS += -lITKWatersheds-4.7
LIBS += -litkzlib-4.7
LIBS += -lITKznz-4.7

win32: PRE_TARGETDEPS += C:/ITK/lib/itksys-4.7.lib
win32: LIBS += "advapi32.lib"
win32: LIBS += "kernel32.lib"
win32: LIBS += "gdi32.lib"
win32: LIBS += "user32.lib"
win32: LIBS += "gdi32.lib"
win32: LIBS += "winspool.lib"
win32: LIBS += "shell32.lib"
win32: LIBS += "Uuid.lib"
win32: LIBS += "ole32.lib"
win32: LIBS += "oleaut32.lib"
win32: LIBS += "comdlg32.lib"
win32: LIBS += "snmpapi.lib"
win32: LIBS += "comctl32.lib"
win32: LIBS += "wsock32.lib"
win32: LIBS += "ws2_32.lib"
win32: LIBS += "rpcrt4.lib"
```