

# Visualização volumétrica de imagens DICOM para iOS

Marcelo da Mata Oliveira

Orientador: Dalton Solano dos Reis

FURB - Universidade Regional de Blumenau  
DSC - Departamento de Sistemas e Computação  
Grupo de Pesquisa em Computação Gráfica, Processamento de  
Imagens e Entretenimento Digital  
[www.inf.furb.br/gcg](http://www.inf.furb.br/gcg)

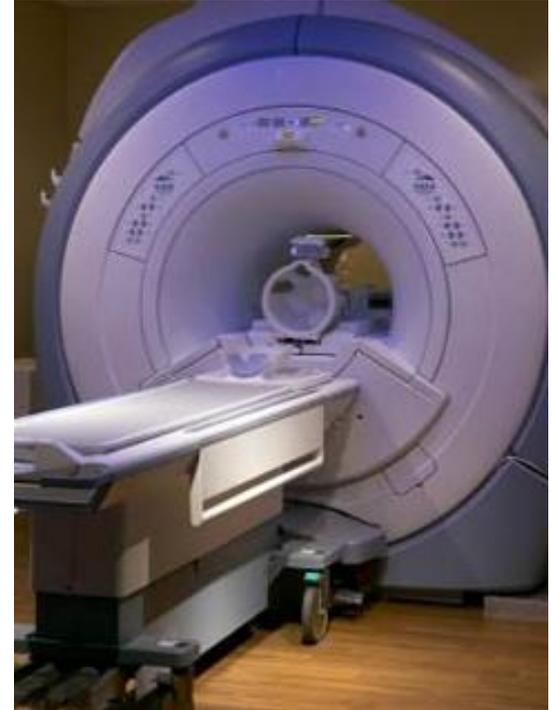


# Roteiro

- Introdução
- Objetivos do trabalho
- Fundamentação teórica
- Desenvolvimento
- Conclusões
- Demonstração

# Introdução

- Evolução computacional
- Computação na medicina
  - exames computadorizados
  - padrão DICOM
  - visualização volumétrica
  - uso de dispositivos móveis



# Objetivos

- ler um arquivo no formato DICOM que se encontra no dispositivo móvel;
- realizar a visualização 2D a partir de imagens adquiridas no arquivo DICOM;
- realizar a visualização 3D a partir de imagens adquiridas no arquivo DICOM.

# Fundamentação teórica

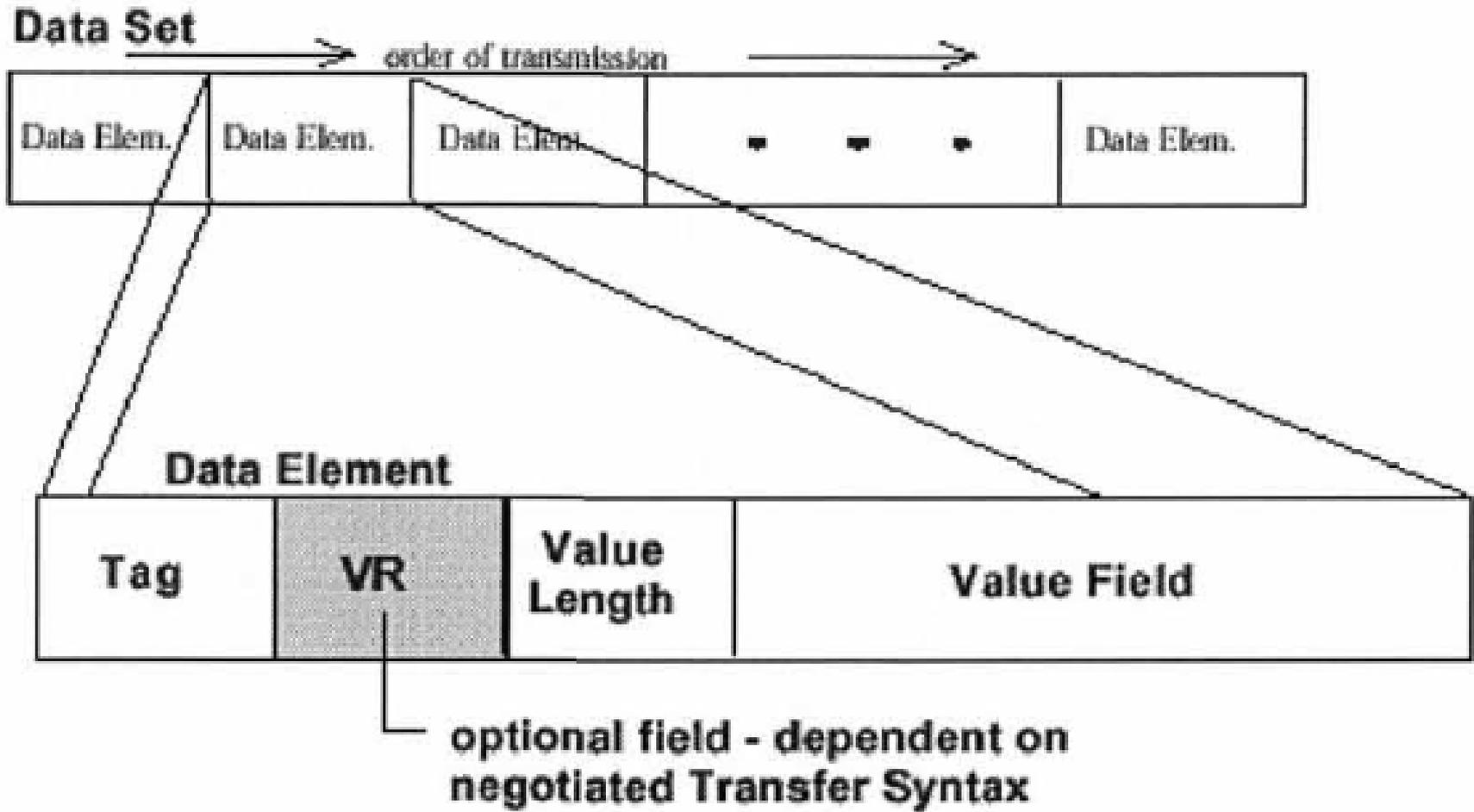
## Padrão DICOM

- Teve seu desenvolvimento iniciado em 1983
- Comitê formado pela American College Radiology (ACR) e National Electrical Manufacturers Associations (NEMA)
- Características

# Padrão DICOM

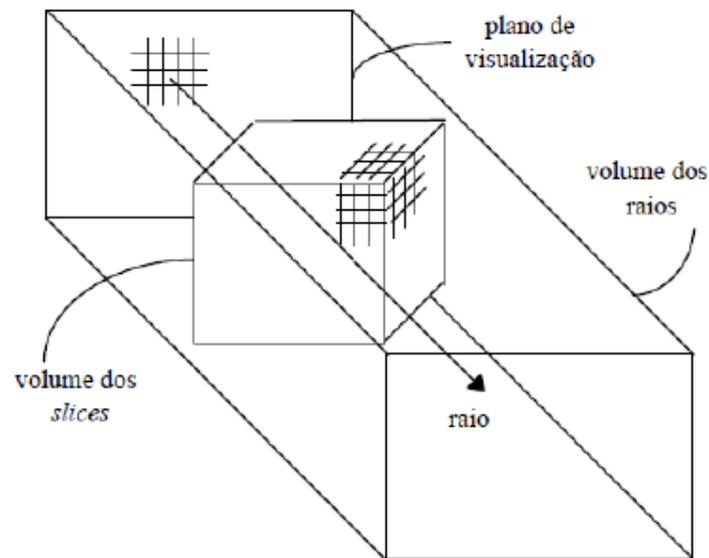
Parte 1: Visão Geral		
Parte 2: Configuração		
Parte 4: Classe de Especificação do Serviço	Parte 3: Definição do Objeto de Informação	Parte 11: Perfil da Aplicação dos Meios de Armazenamento
Parte 5: Estruturas de Dados e Semântica		
Parte 6: Dicionário de Dados		
Parte 7: Troca de Mensagem (operação em rede)	Parte 8: Suporte a Rede para Troca de mensagem (TCP/IP e OSI)	Parte 9: Retirada (Ponto a Ponto)
Parte 10: Meios de Armazenamento e Formato do arquivo para o intercâmbio dos dados		
Parte 12: Formatos dos meios e meios físicos para o intercâmbio dos dados	Parte 13: Retirada	Parte 14: Função de Visualização do Padrão de tons de Cinza.
Parte 15: Perfil de Segurança		

# Padrão DICOM



# Visualização volumétrica

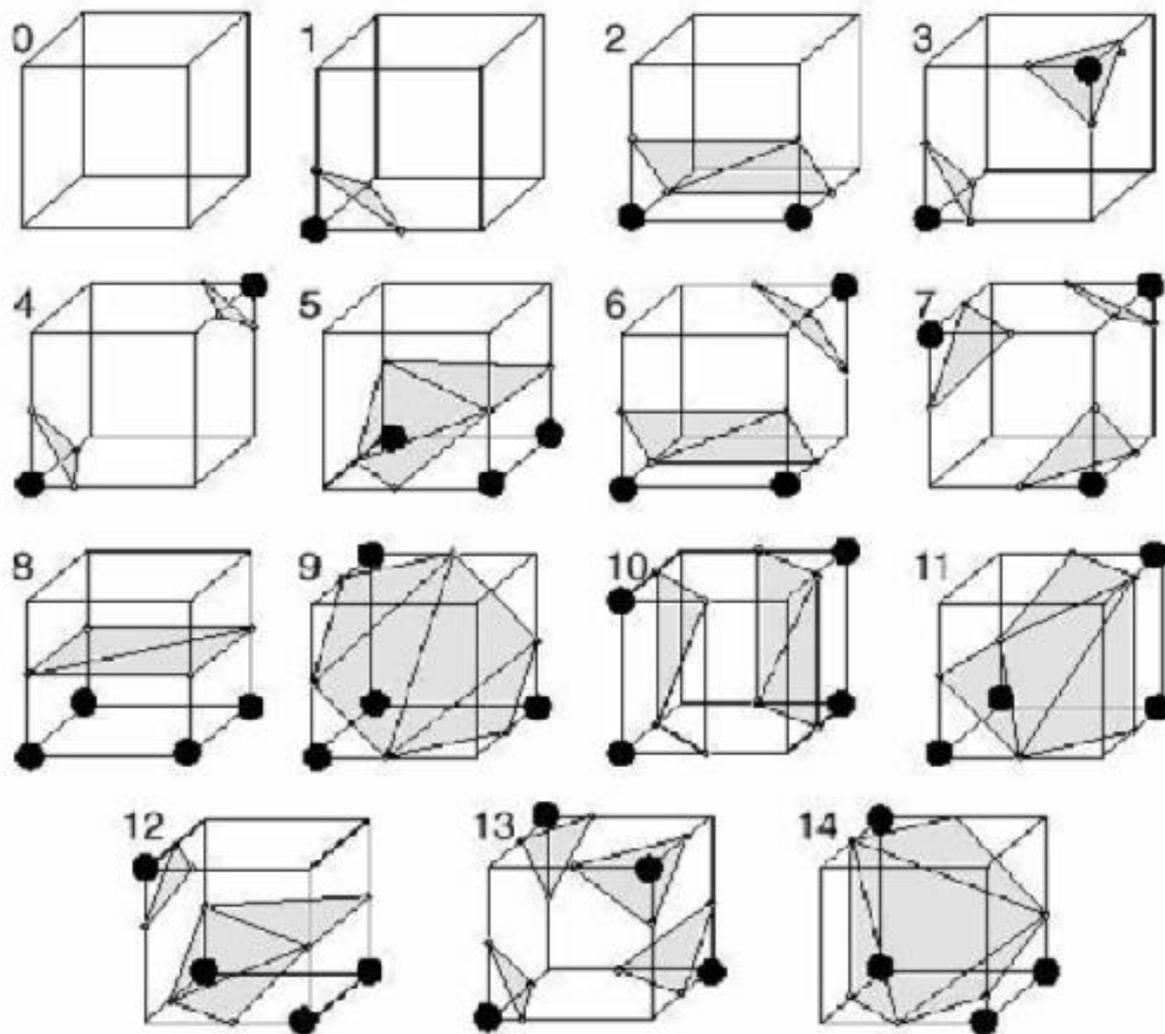
- Visualização direta de volumes
  - sem uso de primitivas geométricas
  - maior qualidade
  - maior custo computacional



# Visualização volumétrica

- Visualização por extração de superfícies
  - detalham a superfície em polígonos
  - menor custo computacional
  - baixa qualidade na visualização de algumas partes do corpo

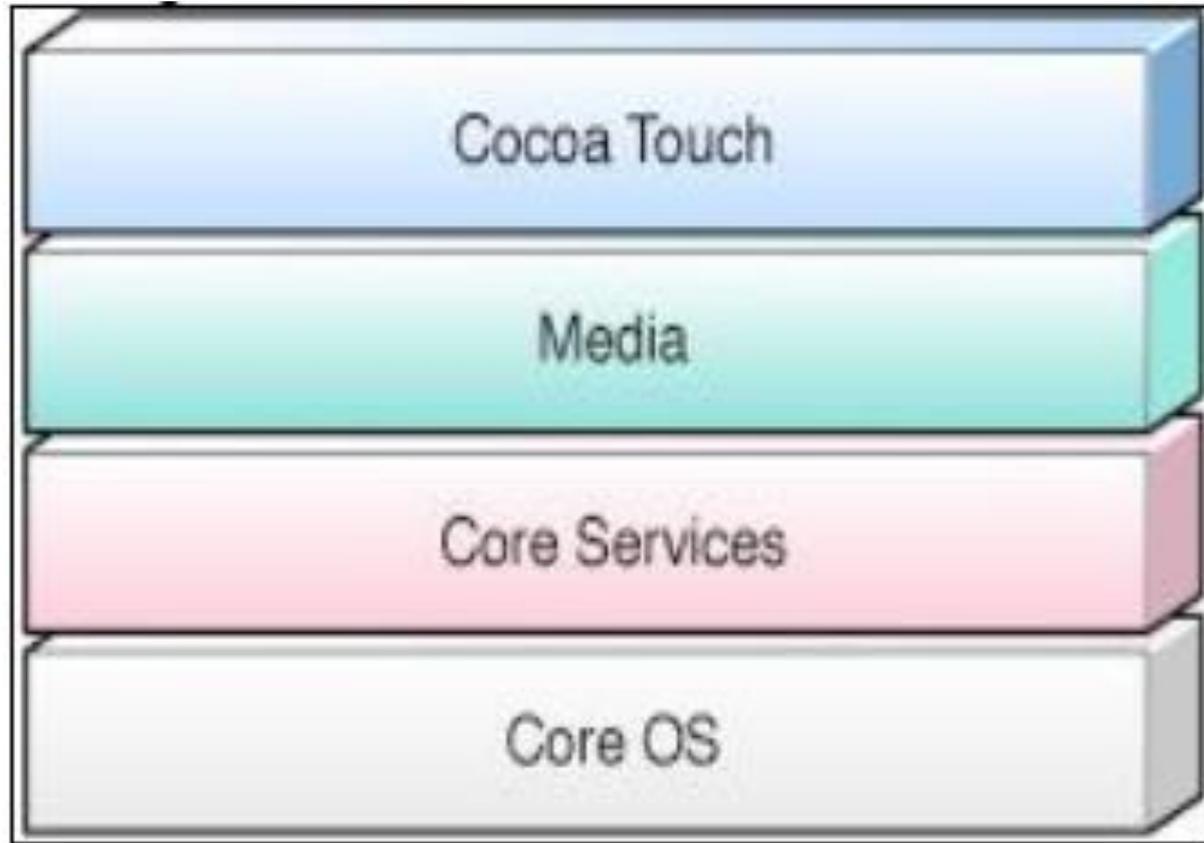
# Visualização volumétrica



# Plataforma iOS

- Sistema operacional iOS para dispositivos móveis da Apple
- Fruto do conhecimento obtido no Mac OS X
- iOS SDK
- Núcleo de controle dividido em 4 camadas

# Plataforma iOS



Fonte: Apple Inc. (2012)

- OpenGL ES

# Trabalhos correlatos

Características\Trabalhos correlatos	InVesalius	Osirix	Roepke (2010)
Ler arquivos DICOM	X	X	
Visualização em duas dimensões	X	X	X
Tratamento das imagens	X	X	X
Visualização volumétrica	X	X	
Visualização interna do volume	X	X	
Visualização em mais de três dimensões		X	
Separação do volume em componentes	X	X	
Fatiamento do volume	X	X	
Geração da imagem em outras direções anatômicas	X	X	
Aplicação para dispositivos móveis		X	X

# Desenvolvimento

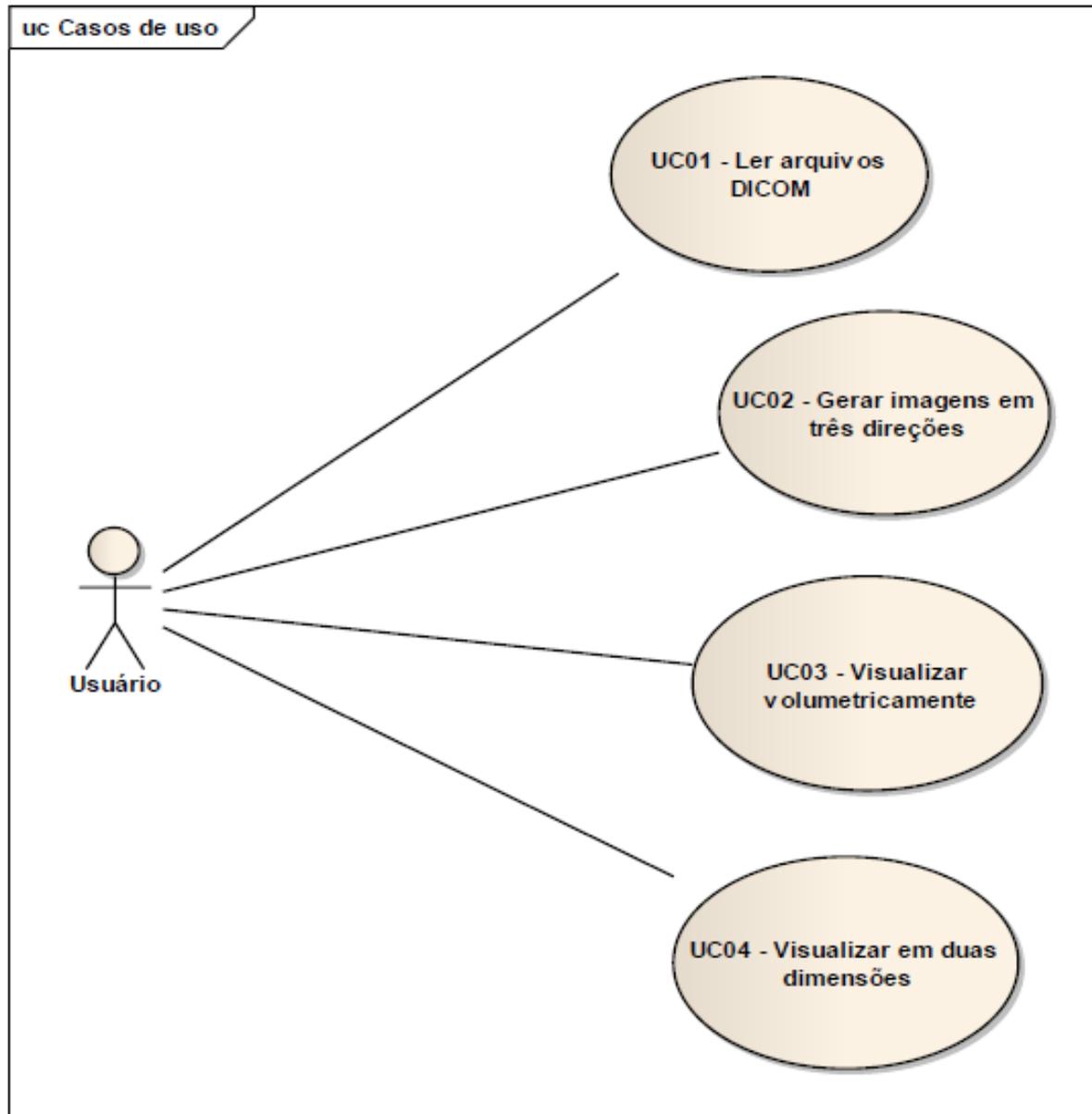
## Requisitos

- Requisitos funcionais
  - ler o cabeçalho e as imagens de um arquivo DICOM
  - apresentar a sequência de imagens em formato 2D contidas no arquivo DICOM na direção anatômica que a imagem foi capturada
  - realizar a visualização volumétrica das imagens DICOM
  - realizar o fatiamento do volume em três direções anatômicas, axial, sagital e coronal

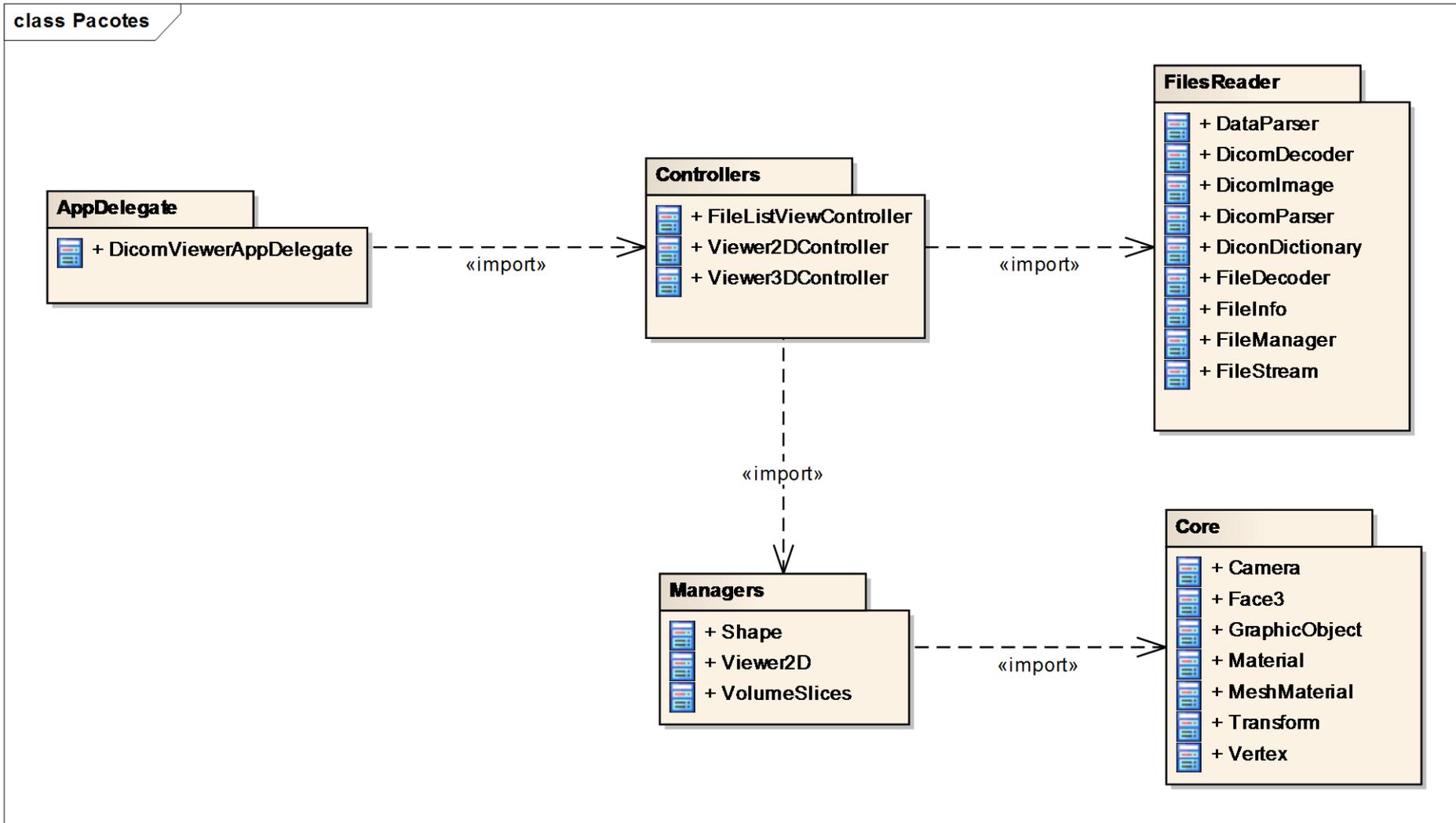
# Requisitos

- Requisitos não-funcionais
  - ser implementado utilizando a linguagem de programação Objective-C
  - ser implementado utilizando o ambiente de desenvolvimento XCode 4
  - ser desenvolvido para executar em dispositivos móveis como o iPhone, iPad e iPod Touch

# Especificação Casos de uso

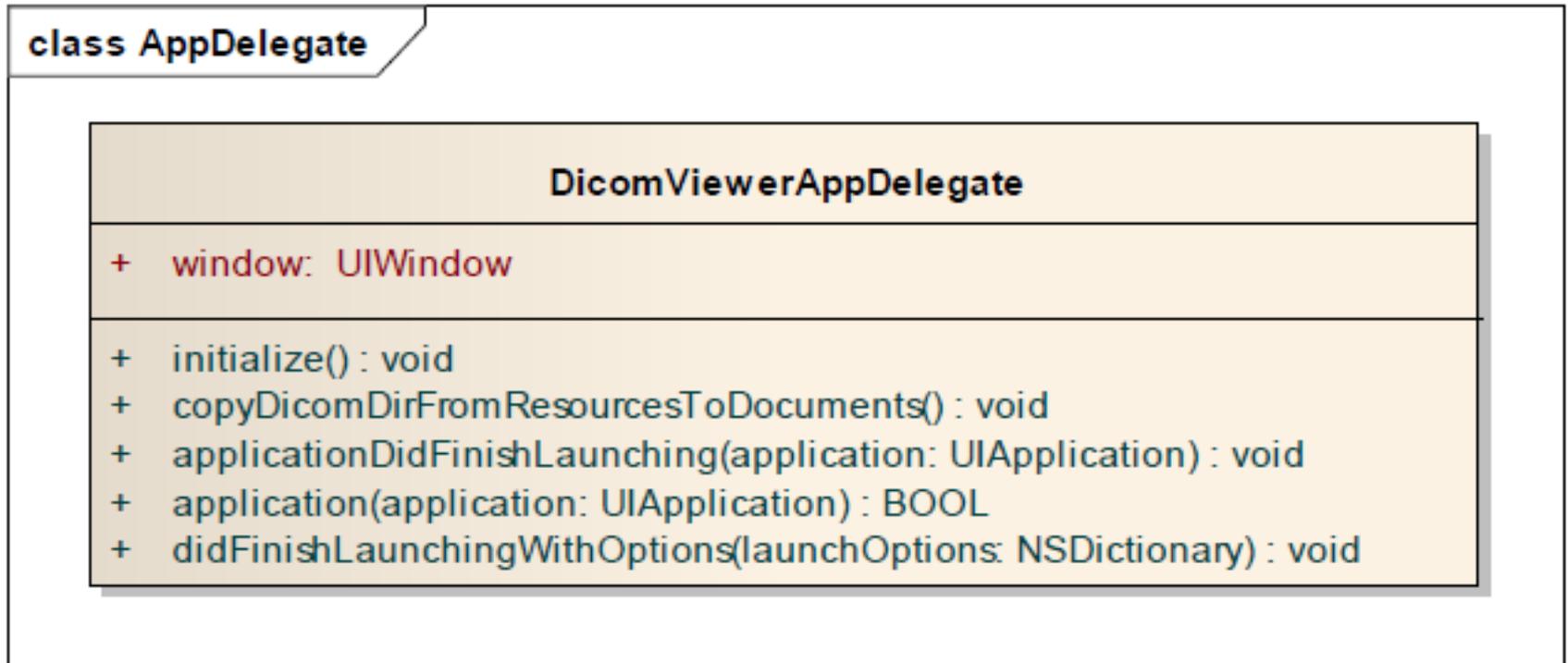


# Diagrama de pacotes



# Diagramas de classes

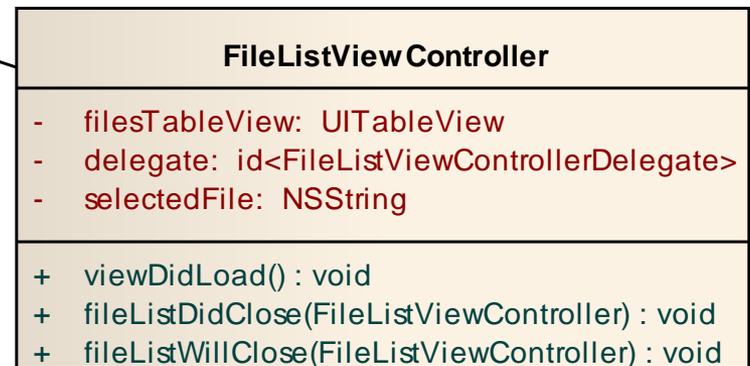
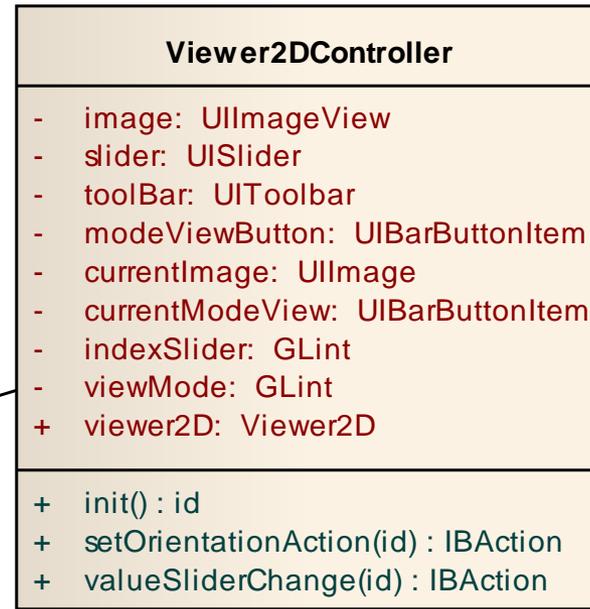
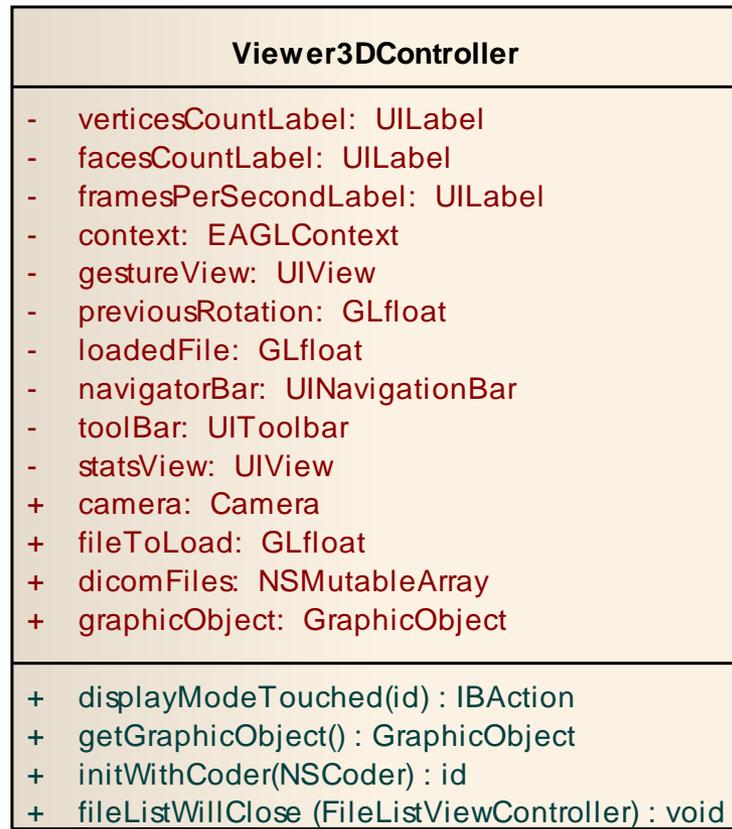
- Pacote AppDelegate



# Diagramas de classes

- Pacote Controllers

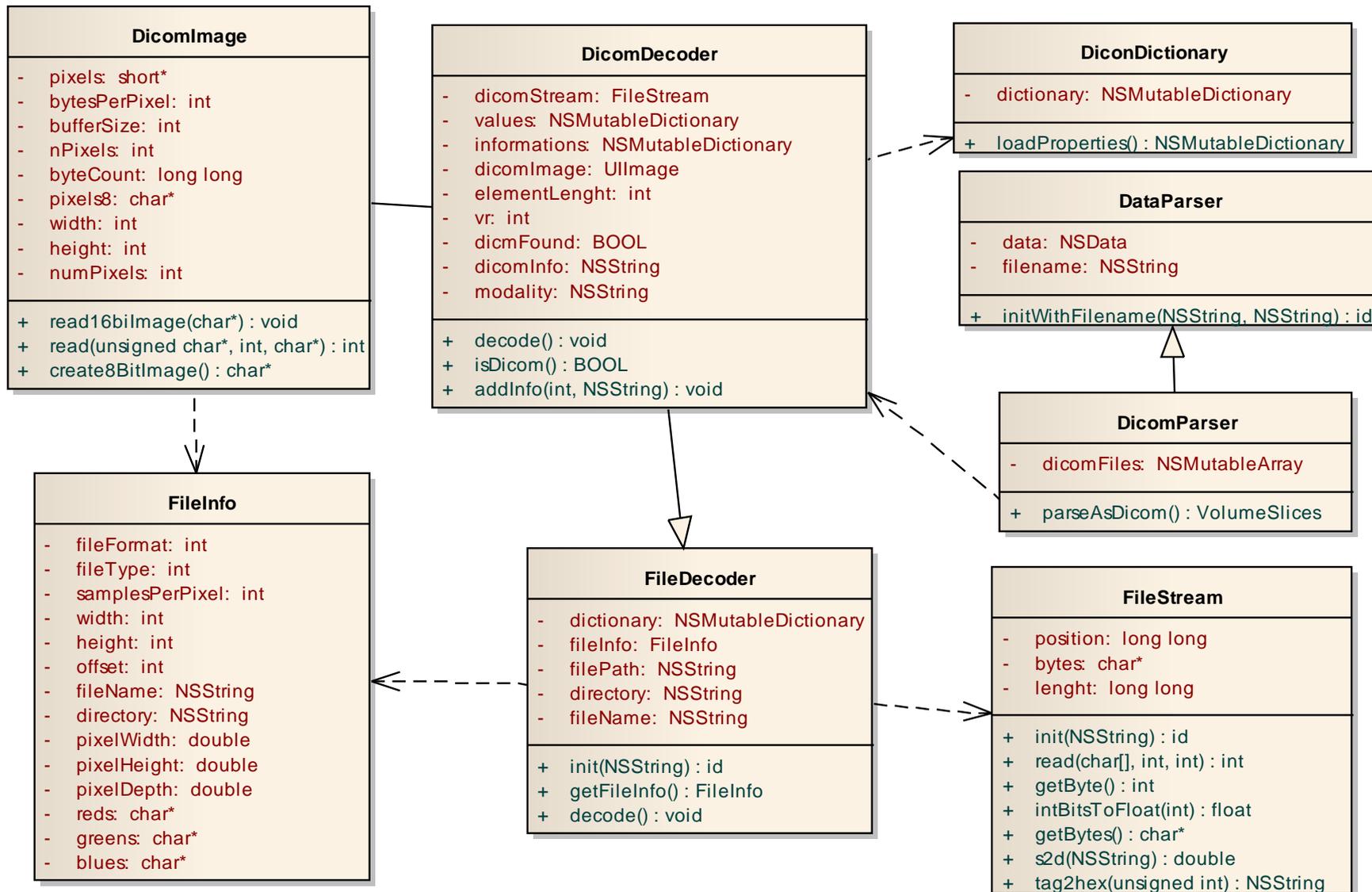
class Controllers



# Diagramas de classes

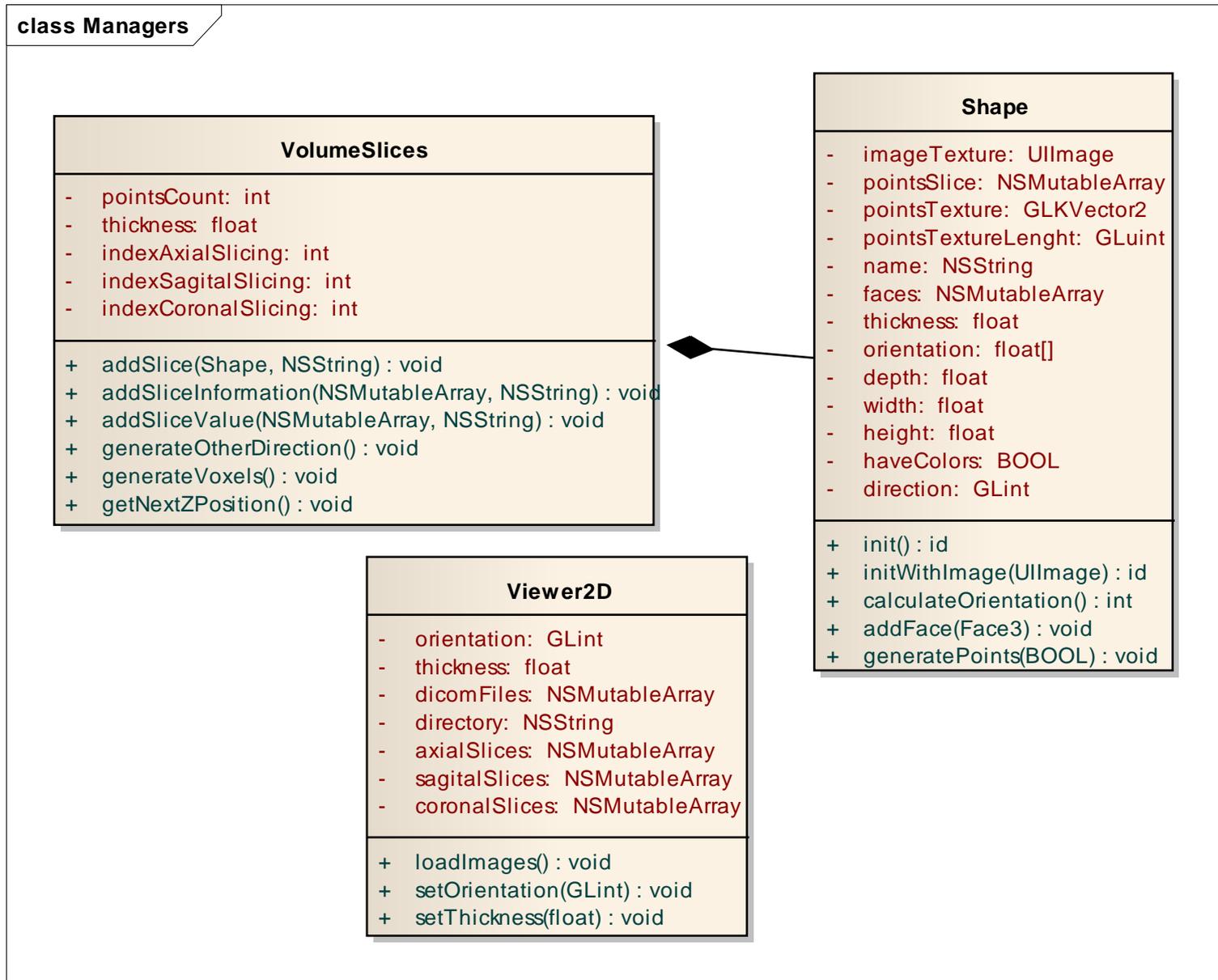
- Pacote FileReader

class FileReader



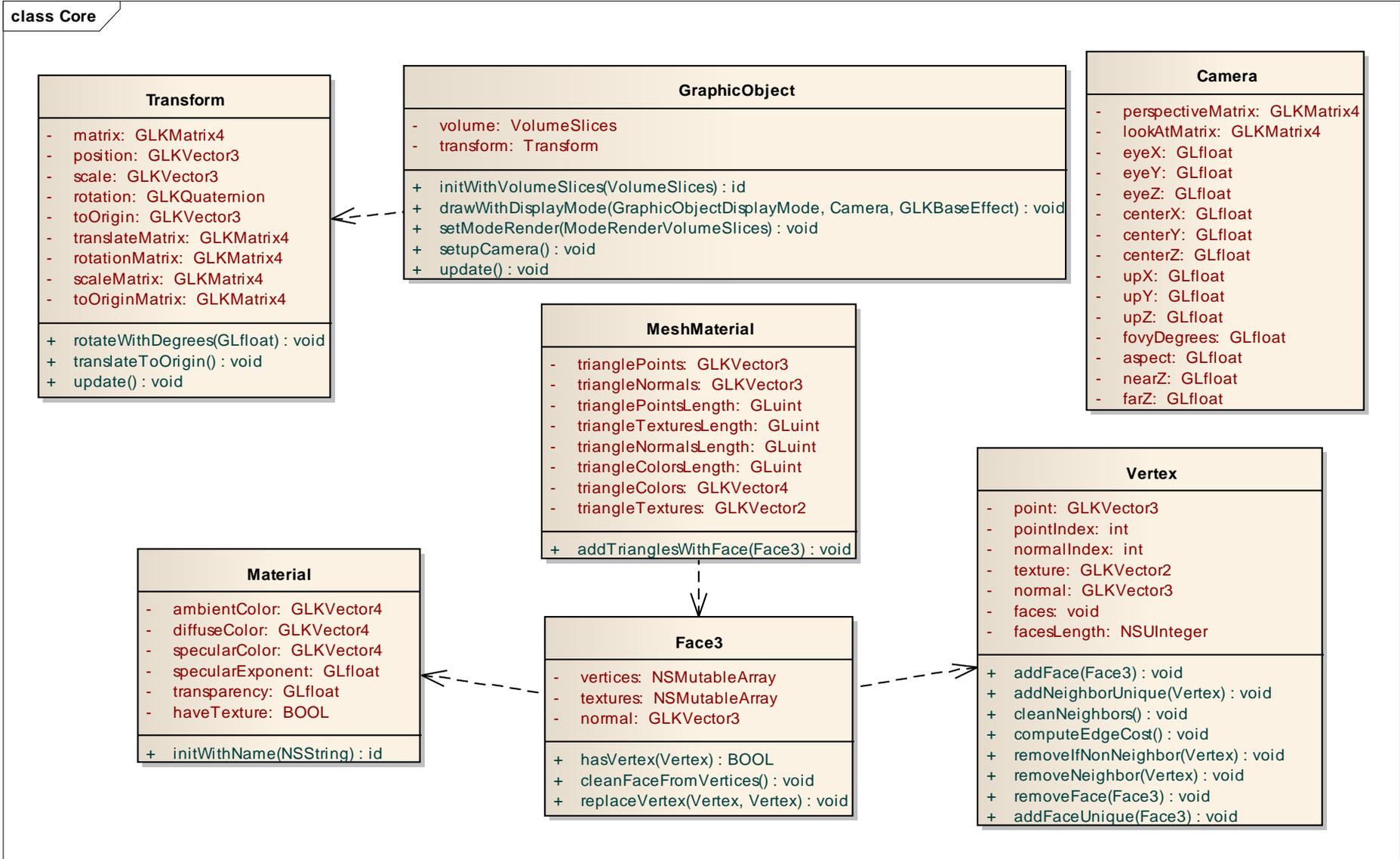
# Diagramas de classes

- Pacote Managers

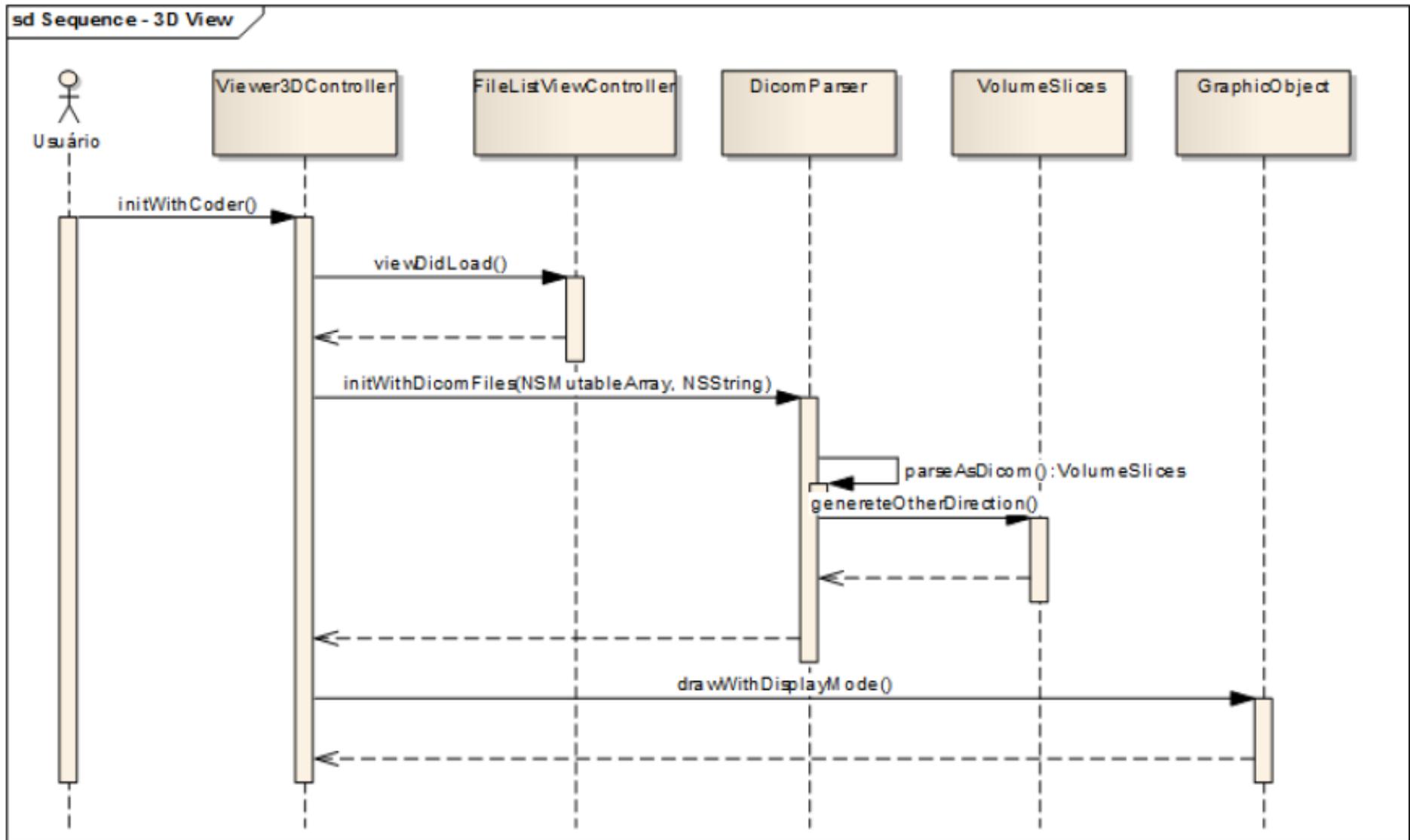


# Diagramas de classes

- Pacote Core



# Diagrama de sequência

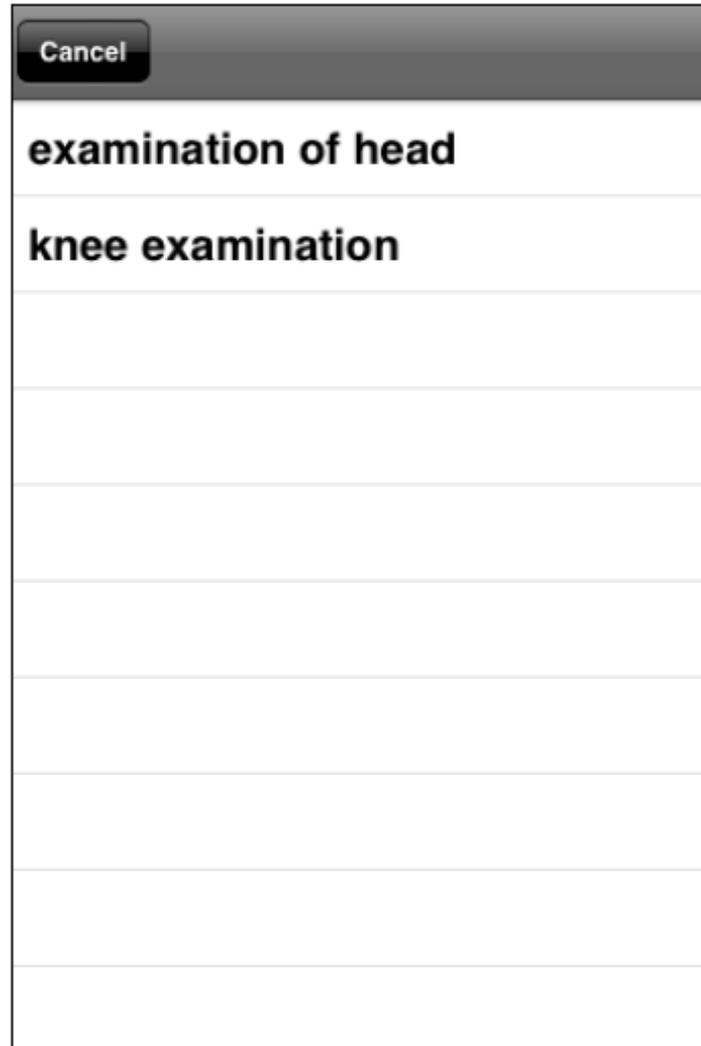


# Implementação

- Técnicas e ferramentas utilizadas
  - Xcode 4.5.1
  - bibliotecas: UIKit; Foundation; GLKit; CoreGraphics; OpenGL ES
- Leitura de arquivos DICOM
  - Baseado no ImageJ
- Visualização volumétrica
  - Adaptação da biblioteca de Imianowsky (2013) para trabalhar com OpenGL ES e eventos de tela

# Operacionalidade

- Escolha do exame



Cancel

**examination of head**

**knee examination**

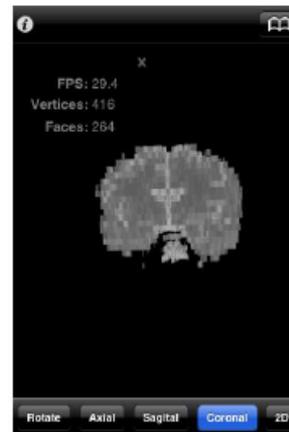
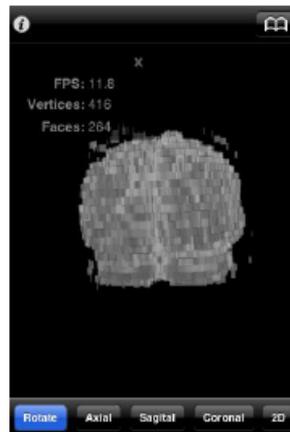
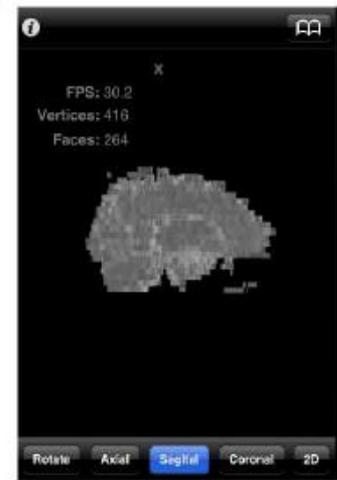
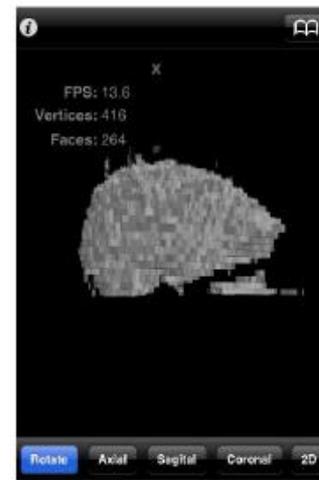
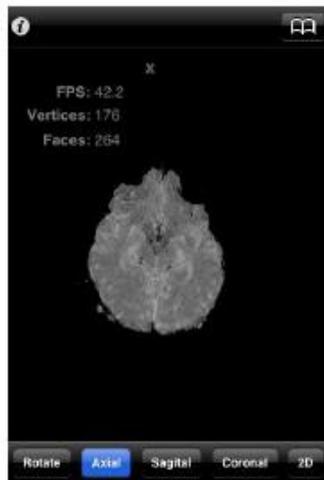
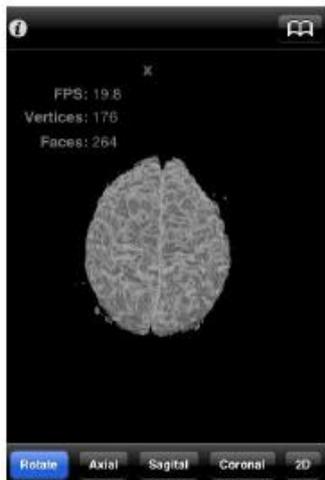
# Operacionalidade

- Visualização volumétrica



# Operacionalidade

- Fatiamento nas 3 direções



# Operacionalidade

- Visualização 2D



# Resultados e discussão

## Testes da geração do volume

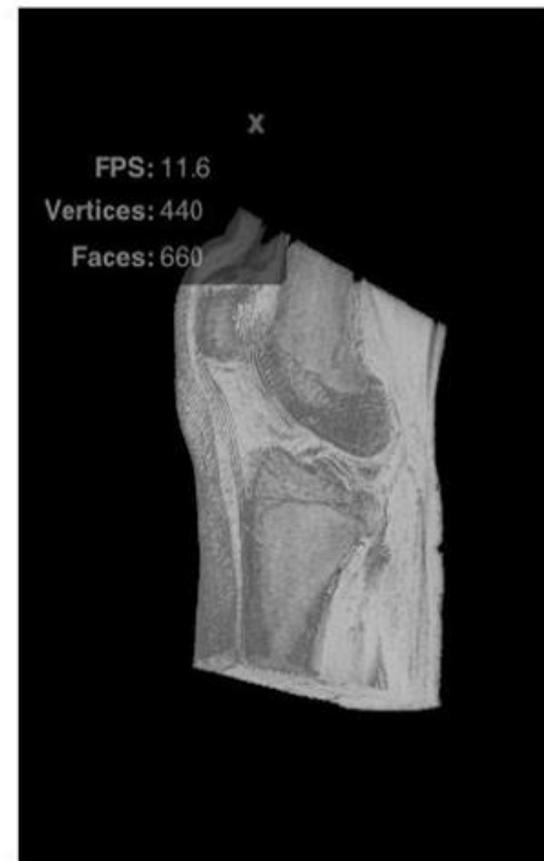
- 2 exames utilizados nos testes, um do crânio e outro do joelho
- Exame do crânio
  - imagens com 256x256
  - 22 fatias
- Exame do joelho
  - imagens com 512x512
  - 55 fatias

# Testes da geração do volume

- Leitura dos arquivos DICOM
  - Implementação de uma biblioteca para ler os arquivos DICOM em iOS
  - Biblioteca lê somente arquivos com imagens de 16 bits com e sem sinal
- Geração do volume
  - cada fatia com uma imagem como textura
  - definição de um limiar para determinar que pixel representa uma parte do corpo
  - exclusão dos pixels que não fazem parte do corpo pela manipulação da opacidade

# Testes da geração do volume

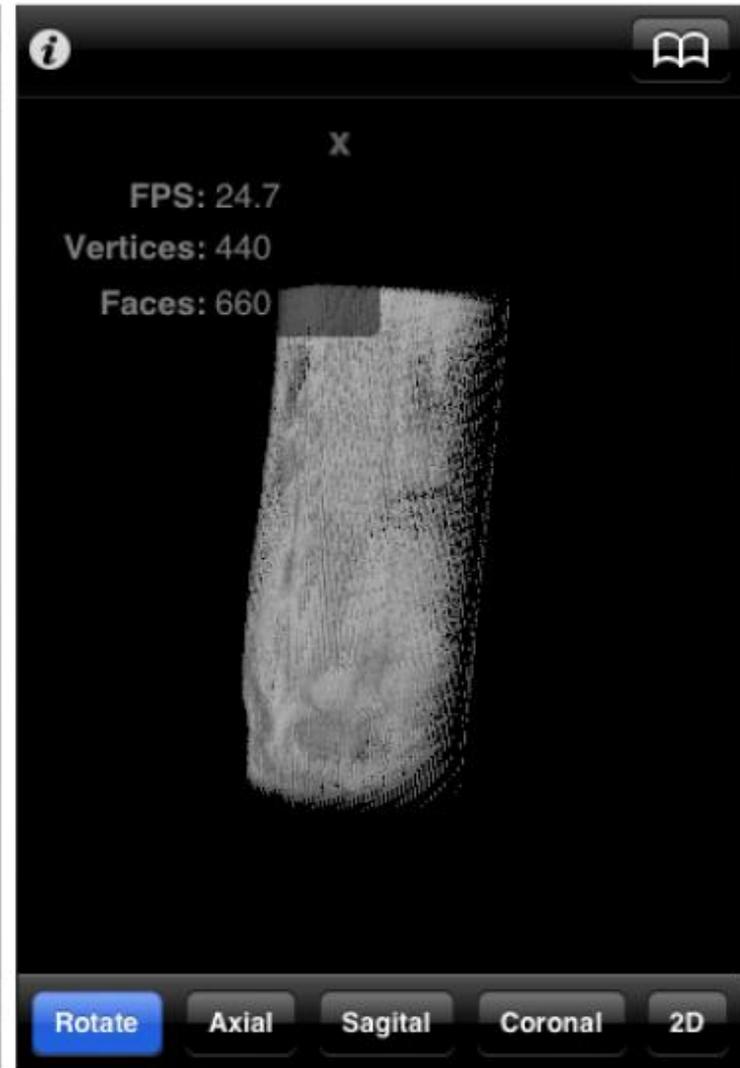
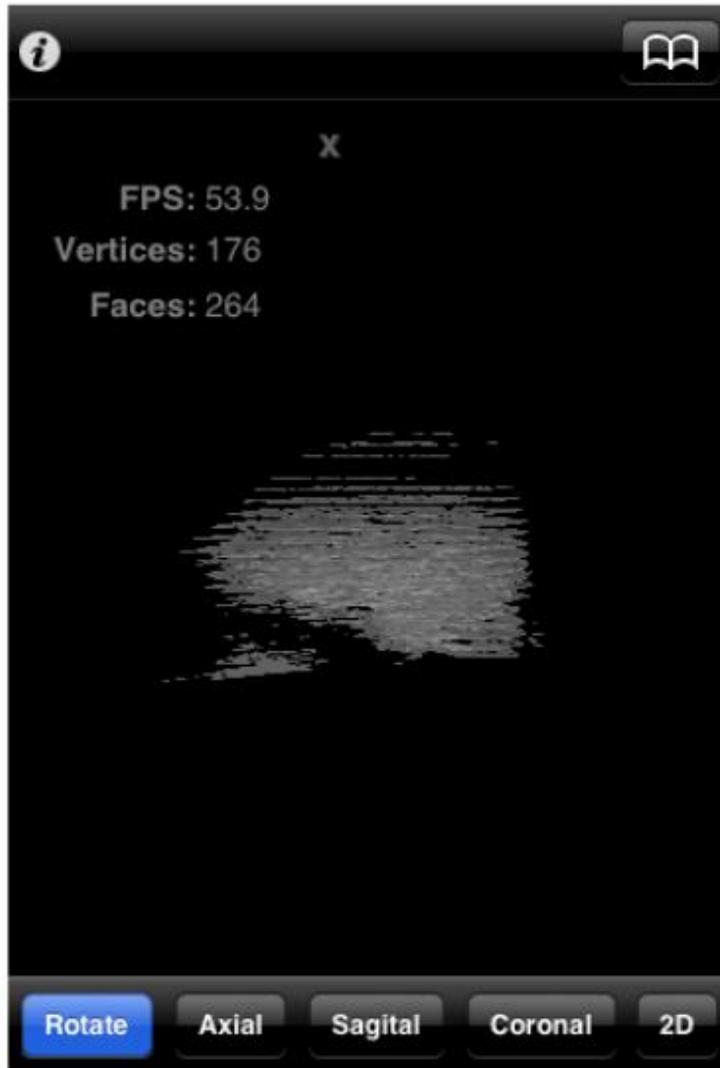
Ruídos



# Testes da geração do volume

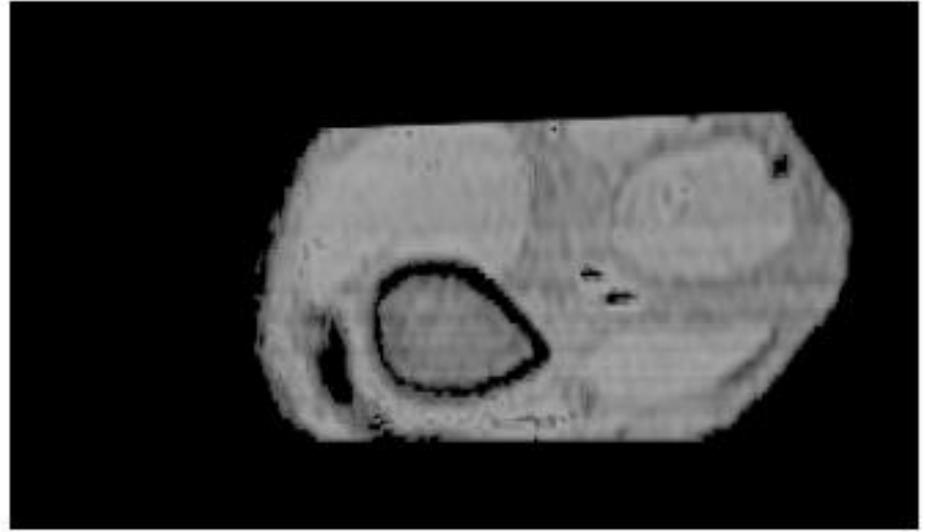
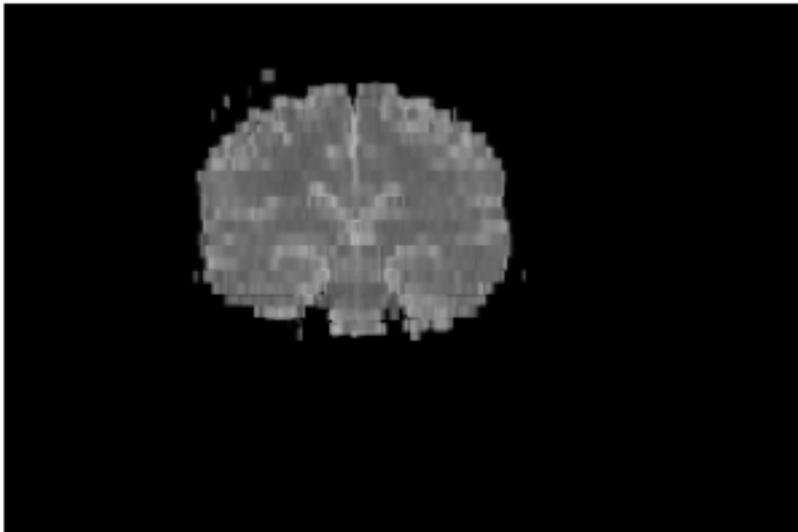
- distanciamento entre as fatias pelo valor da tag  
spacing between slices
- espaço entre as fatias é visível ao usuário

# Testes da geração do volume



# Testes da geração do volume

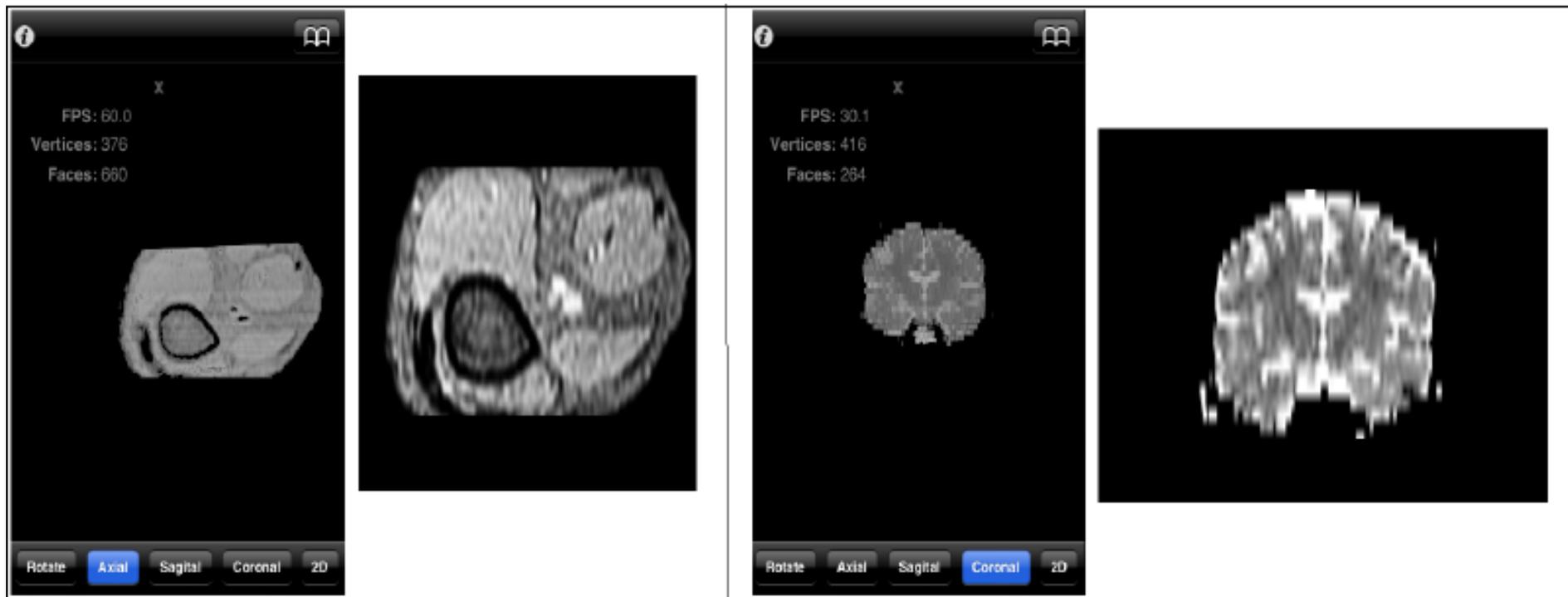
- Fatiamento do volume
  - geração das imagens nas outras direções
  - quanto maior a quantidade de fatias melhor a resolução da imagem gerada



# Comparação com trabalhos correlatos

Características\Trabalhos correlatos	InVesalius	Osirix	Roepke (2010)	Oliveira (2013)
Ler arquivos DICOM	X	X		X
Visualização em duas dimensões	X	X	X	X
Tratamento das imagens	X	X	X	
Visualização volumétrica	X	X		X
Visualização interna do volume	X	X		
Visualização em mais de três dimensões		X		
Separação do volume em componentes	X	X		
Fatiamento do volume	X	X		X
Geração da imagem em mais de uma direção anatômica	X	X		X
Aplicação para dispositivos móveis		X	X	X

# Comparação com trabalhos correlatos



# Consumo de memória e desempenho

- Foi utilizado nos testes um iPad4
  - 1GB de memória DDR2 RAM
  - processador dual core Apple Swift de 1.4 GHz
  - processador gráfico Quad core PowerVR SGX554MP4
- Programa Instruments do Xcode para obter os dados para a análise

# Consumo de memória

- Visualização do exame do crânio no iPad4

Momento	Uso inicial de memória	Uso mínimo de memória	Uso máximo de memória
Antes da seleção do exame	62,98MB	32,91MB	62,98MB
Ao ler os arquivos do exame	33,03MB	33,03MB	86,79MB
Visualização volumétrica do exame	43,46MB	43,46MB	43,46MB

- Visualização do exame do joelho no iPad4

Momento	Uso inicial de memória	Uso mínimo de memória	Uso máximo de memória
Antes da seleção do exame	62,98MB	32,91MB	62,98MB
Ao ler os arquivos do exame	49,87MB	49,87MB	324,11MB
Visualização volumétrica do exame	131,07MB	131,07MB	131,03MB

# Consumo de memória

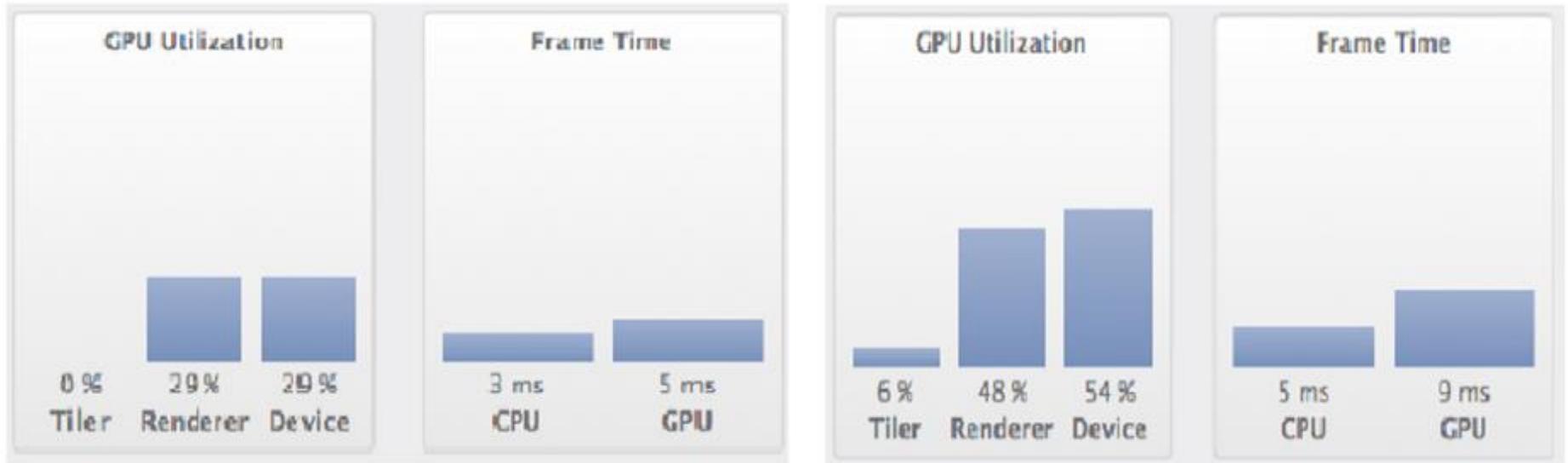
- Uso de memória maior que o limite do dispositivo na geração das imagens
- Teste da geração das imagens no simulador

Exame	Uso inicial de memória	Uso máximo de memória
Crânio	64,06MB	87,48MB
Joelho	203,03MB	315,67MB

# Desempenho

Ação	Tempo (segundos) do exame do crânio	Tempo (segundos) do exame do joelho
Ler os arquivos do exame e desenhar o volume	5,618	14,207
Gerar as imagens (simulador)	0,758	3,07

- Visualização 3D do crânio e do joelho



# Conclusões

- Atende aos requisitos propostos
- Contribuição com uma biblioteca que lê arquivos DICOM no iOS
- Fatiamento do volume em três direções anatômicas
- Geração de imagem com resultados próximos ao do InVesalius
- Aplicação desenvolvida para iOS

# Conclusões

- Limitações
  - Impossibilidade de visualizar elementos do corpo separados
  - Impossibilidade de realizar o fatiamento em outras direções anatômicas
  - Espaço visível ao usuário entre uma fatia e outra
  - Imagens com baixa resolução quando o exame tem poucas fatias
  - Impossibilidade de gerar as imagens no dispositivo

# Extensões

- Extender a biblioteca para ler arquivos DICOM com imagens de qualquer tamanho
- Otimizar o uso de memória para utilizar todos os recursos da aplicação no dispositivo
- Renderizar o contorno do volume
- Implementar o algoritmo ray casting
- Implementar o algoritmo marching cubes
- Renderizar elementos diferentes do corpo
- Realizar o fatiamento em mais de três direções
- Testar o limite dos dispositivos móveis com a execução destes algoritmos

Demonstração

# Fim

- Obrigado!