Visualizador de imagens radiológicas 2D para iPhone

Acadêmico: Marwin Roepke
Orientador: Dalton Solano dos Reis

ROTEIRO

- Introdução / Objetivos
- Fundamentação teórica
  - Tecnologias
  - Trabalhos correlatos
- Desenvolvimento
  - Principais requisitos
  - Implementação
  - Resultados e discussão
- Conclusão / Extensões

INTRODUÇÃO

- Estudar o padrão de imagens radiológicas no padrão DICOM
- Imagens radiológicas são geradas pelos tomógrafos
- Imagens JPEG obtidas das imagens DICOM
- Radiografias em um iPhone, ajudando no prontuário

OBJETIVOS DO TRABALHO

- Visualizar imagens radiológicas 2D em iPhone
- Efetuar alteração de zoom, brilho e contraste nas imagens apresentadas, bem como a marcação de regiões da imagem
- Buscar as imagens através da rede 3G e wireless

FUNDAMENTAÇÃO TEÓRICA

LINGUAGEM DE PROGRAMAÇÃO PARA IPHONE

- Lançamento no ano de 2007
- Criada por Brad Cox na StepStone Corporation
- Caracterizada conjunto de extensões do padrão de linguagem ANSI C
- Baseada no Smalltalk

IMAGENS RADIOLÓGICAS

- Feixes em forma de leques que atravessam uma seção
- Recebido por arranjo circular de detectores
- Calculado um valor dentro de uma escala de tons cinza para cada pixel do corte
- As imagens são geradas para o padrão DICOM desde 1985
- Padronizadas pela NEMA e pela ACR

VISUALIZAÇÃO DE IMAGENS - JPEG

- Os objetos 3D tratados considerar como fossem objetos 2D
- Imagem 3D utilizando várias imagens 2D, como JPEG
- JPEG compressão de imagens existam perdas de informações toleráveis

TRABALHOS CORRELATOS

CONVERSÃO DE IMAGENS DO FORMATO DICOM VISANDO A INTER-OPERACIONALIDADE DE SISTEMAS ATRAVÉS DA WEB

- Visualizar as imagens radiológicas num *browser*
- Não houve o atendimento da flexibilidade exigida para a utilização em prontuários

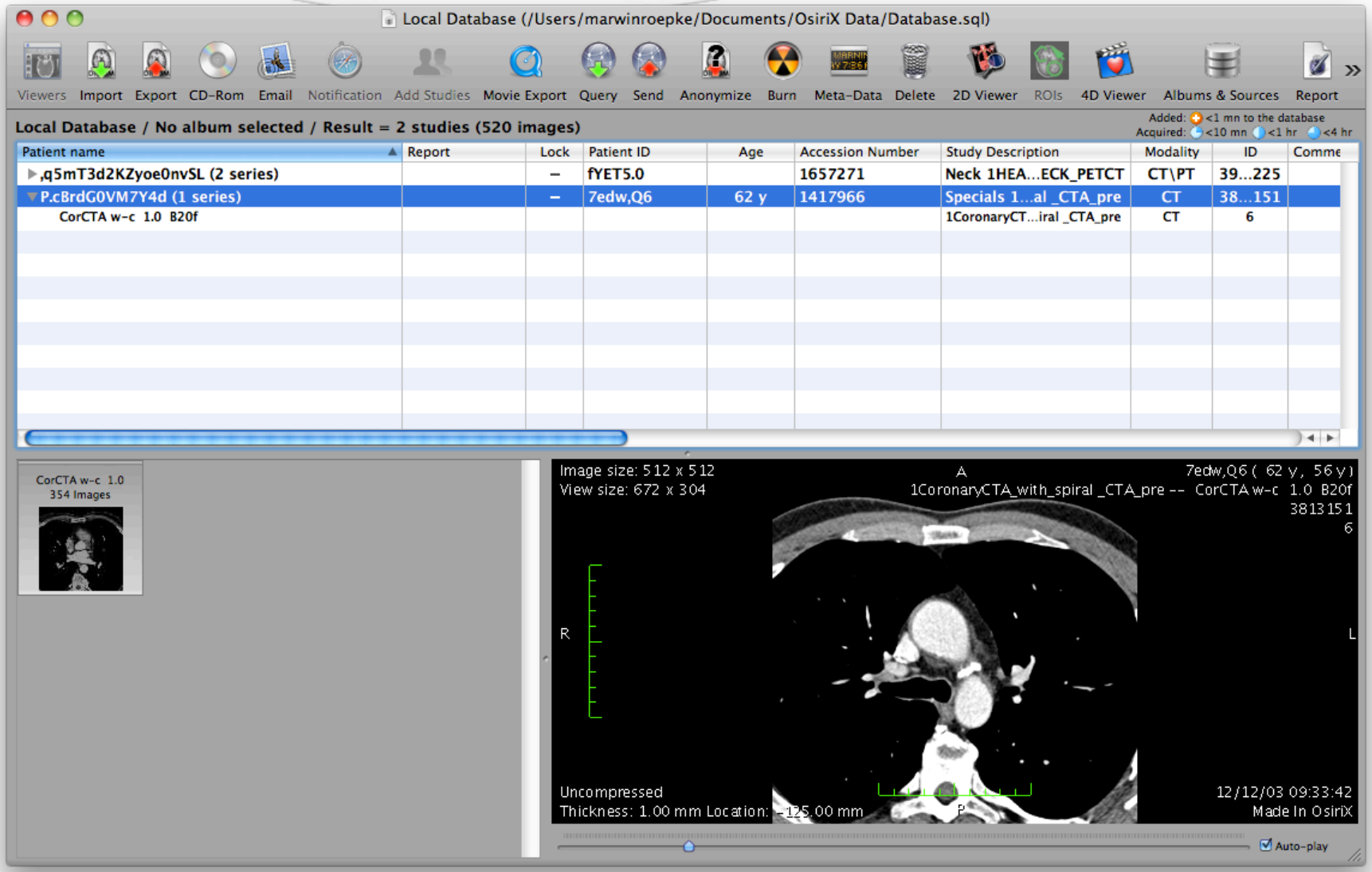
ANÁLISE DE IMAGENS RADIOLÓGICAS VIA DISPOSITIVOS MÓVEIS

- Um sistema distribuído de análises de imagens médicas para a web, utilizando *web service*
- Implementa a conversão das imagens de DICOM para JPEG e clientes para PDA e celular

OSIRIX

- Aplicação para estações Mac OS X, código livre
- Permite a navegação e visualização de imagens
- Modos visualizar em: 2D, 3D, 4D e 5D
- Em novembro de 2008 conta com uma versão para o iPhone paga.

Osirix Mac OS X



Osirix iPhone

DESENVOLVIMENTO

REQUISITOS

- O sistema deverá utilizar o multitoque para dar *zoom* as imagens (RF)
- O sistema deverá permitir adicionar brilho e contraste as imagens (RF)
- O sistema deverá permitir inserir marcações na imagem (RF)
- O sistema deverá utilizar o acelerômetro para retirar as marcações na tela (RF)
- O sistema deverá acessar imagens utilizando a rede 3G e/ou wireless (RNF)
- O sistema deverá ser implementado na linguagem Objective-C (RNF)

DIAGRAMA DE CASOS DE USO

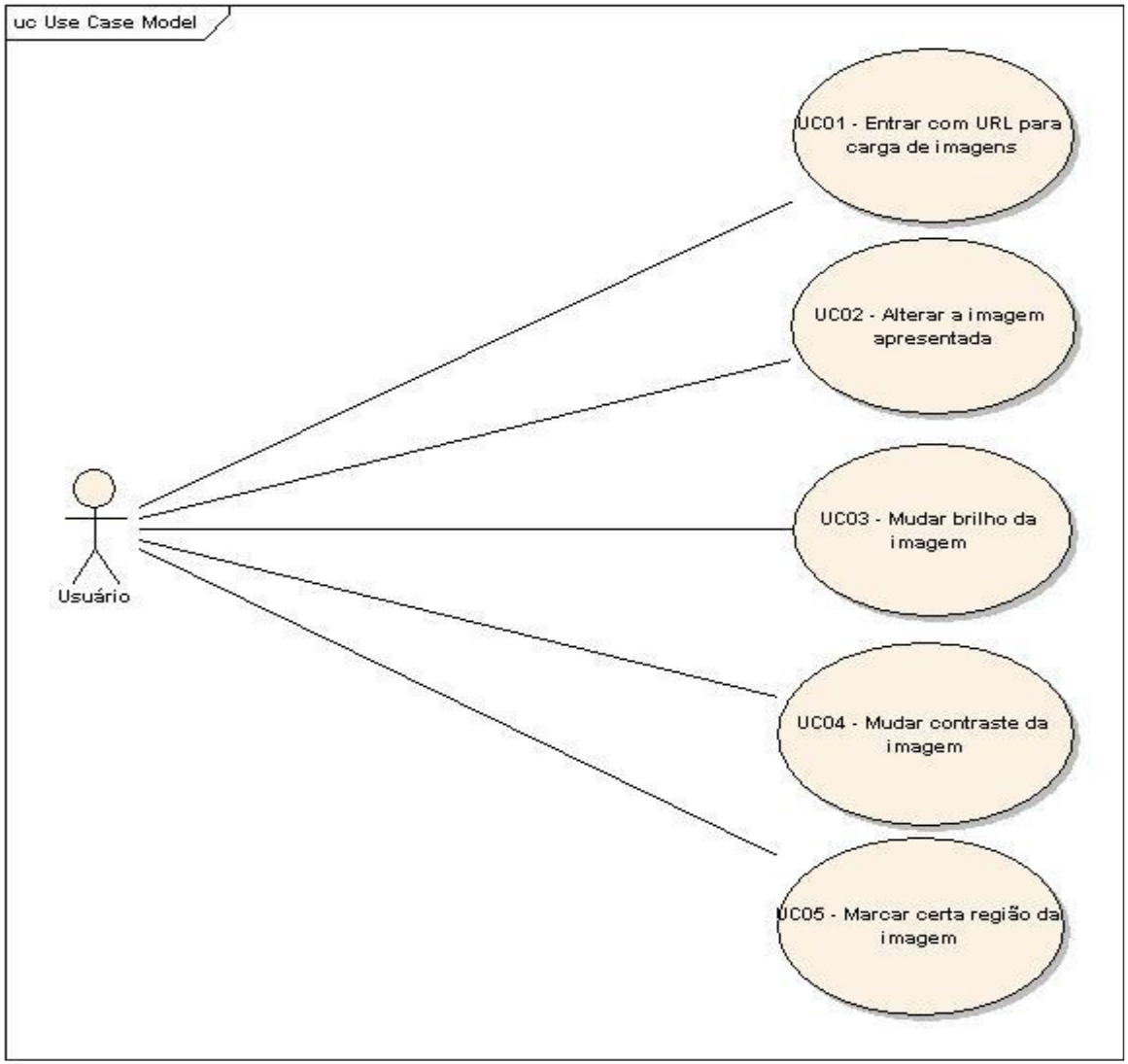
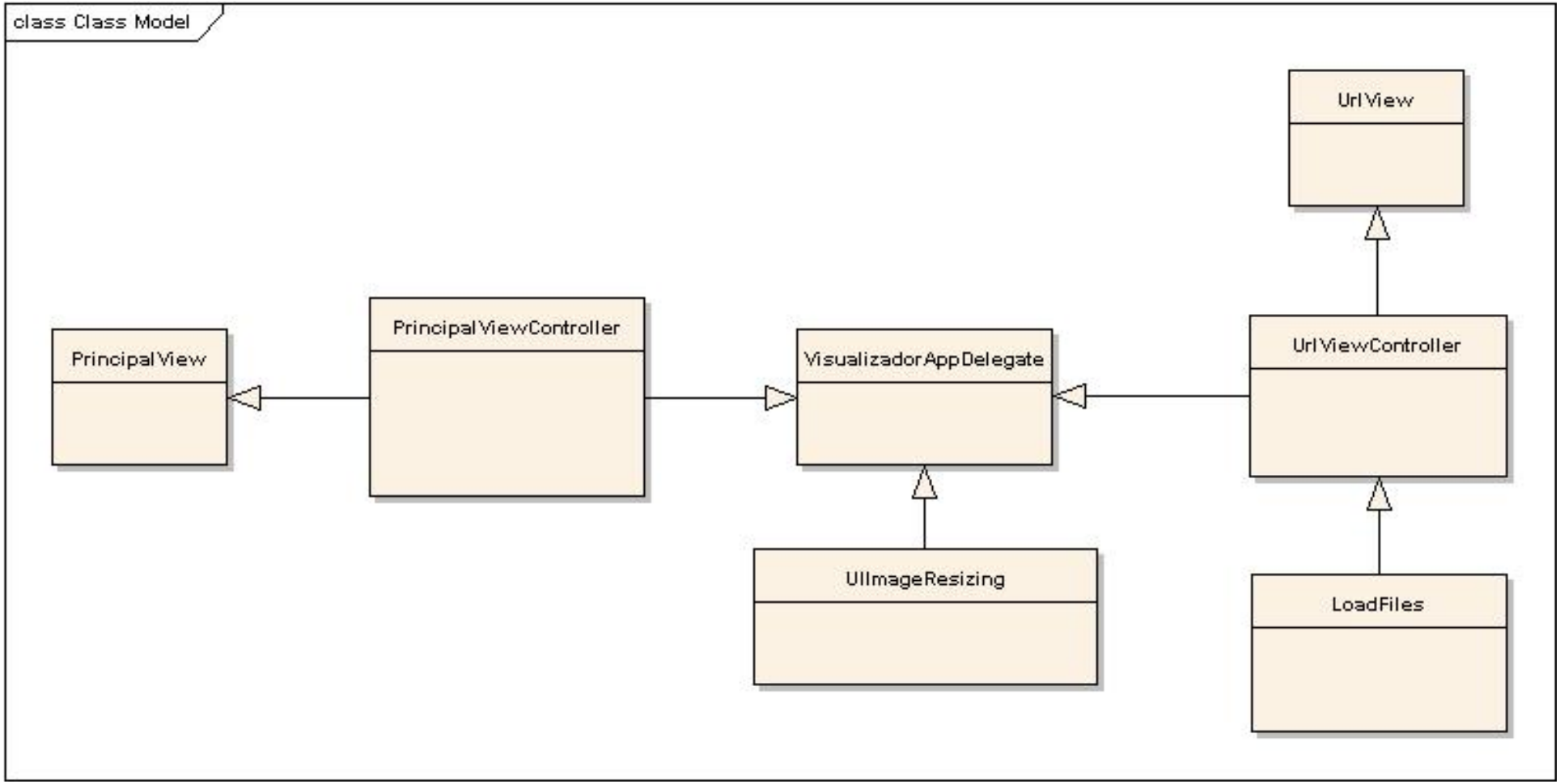
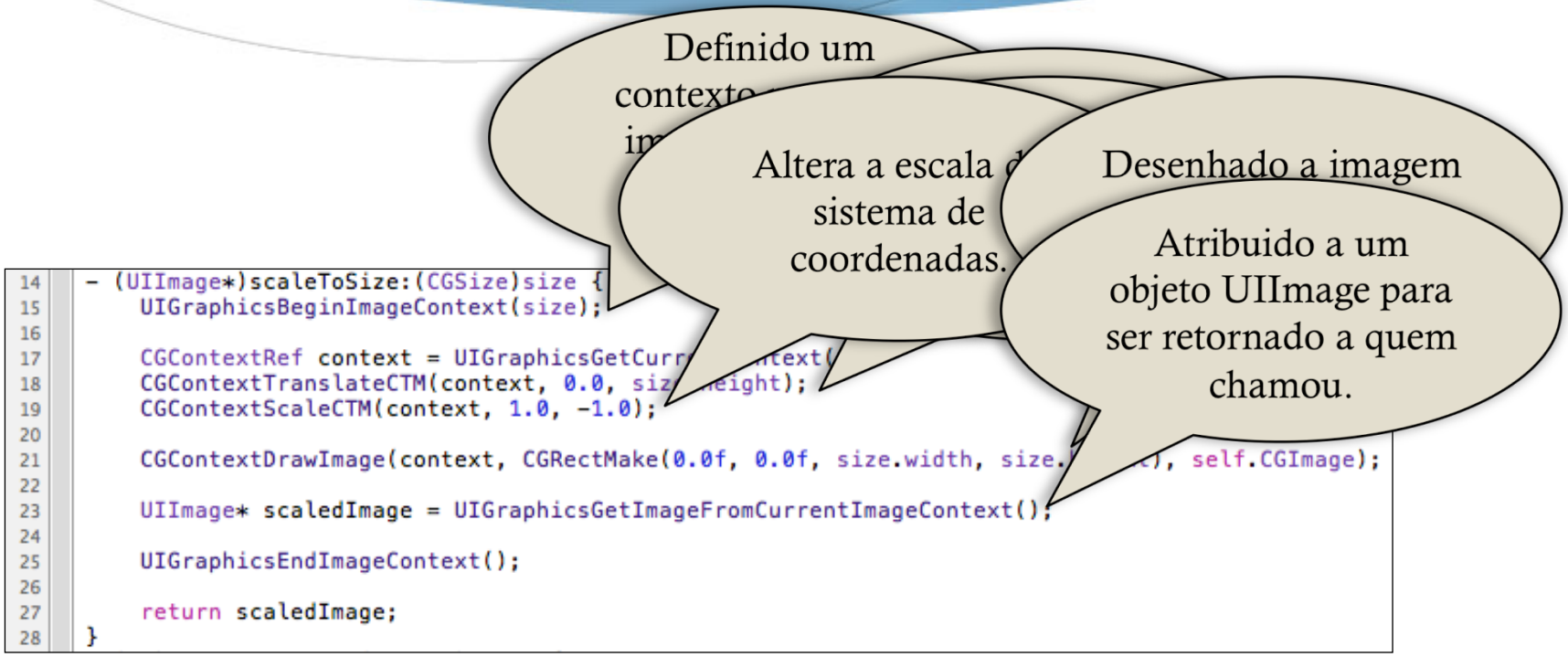


DIAGRAMA DE CLASSES



IMPLEMENTAÇÃO - AJUSTE DE ESCALA DAS IMAGENS




```
14  - (UIImage*)scaleToSize:(CGSize)size {
15      UIGraphicsBeginImageContext(size);
16
17      CGContextRef context = UIGraphicsGetCurr
18      CGContextTranslateCTM(context, 0.0, siz
19      CGContextScaleCTM(context, 1.0, -1.0);
20
21      CGContextDrawImage(context, CGRectMake(0.0f, 0.0f, size.width, size.
22
23      UIImage* scaledImage = UIGraphicsGetImageFromCurrentImageContext();
24
25      UIGraphicsEndImageContext();
26
27      return scaledImage;
28  }
```

IMPLEMENTAÇÃO - TRANSIÇÃO DAS IMAGENS

```
245  -(void)performTransition
246  {
247      // Primeiro crie um objeto CATrans
248      CATransition *transition = [C
249      // Anima por 1 segundo
250      transition.duration = 1;
251      // usando a função timi
252      transition.timingFuncti                                    TimingFunctionDefault];
253
254      // Seta o tipo de transiç
255      transition.type = kCATransi
256
257      // Finalmente, para evitar a                                nós atribuímos a nós mesmos a delegaçao para a animação e
             esperamos a
258      //-AnimationDidStop:fi                                     flag já não esta em transição.
259      transitioning = YES
260      transition.delega
261
262      // Em seguida ad                                           executar a transição com base em como podemos alterar o
             conteúdo.
263      [containerView.lay
264
265      // Aqui nós temos a vie                                    com que o Core Animation para animar view1 um hora
             e outra a View2
266
267      view01.hidden = YES;
268      view02.hidden = NO;
269
270      // E assim que vamos continu                               rocar entre as duas imagens, trocamos as variáveis de instância.
271      UIImageView *tmp = view02;
272      view02 = view01;
273      view01 = tmp;
274  }
```

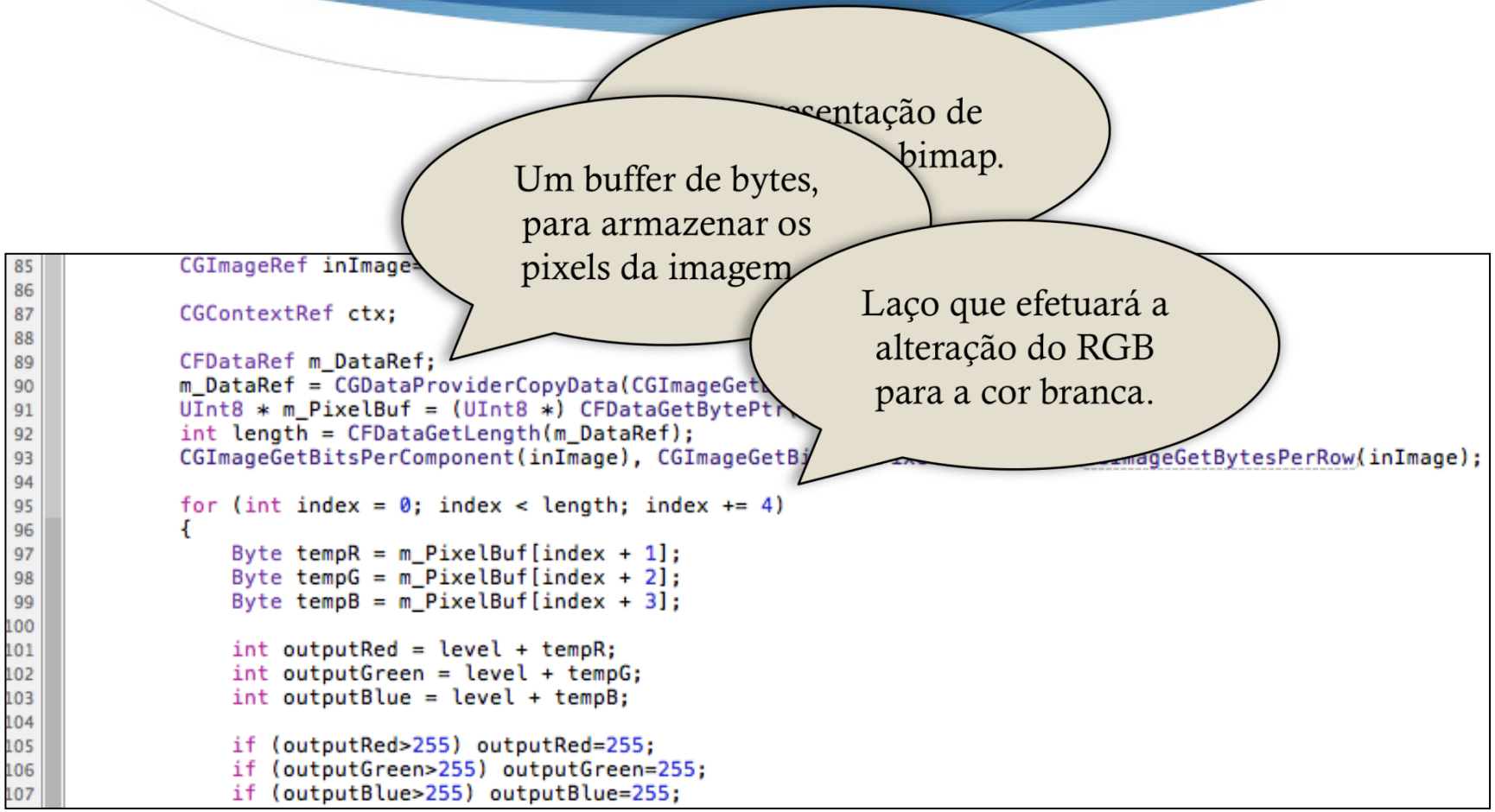
Instanciado um objeto para efetuar as transições.

Definimos a delegação do objeto para controlar a animação

Definimos qual objeto é

Efetua a troca das imagens nas views

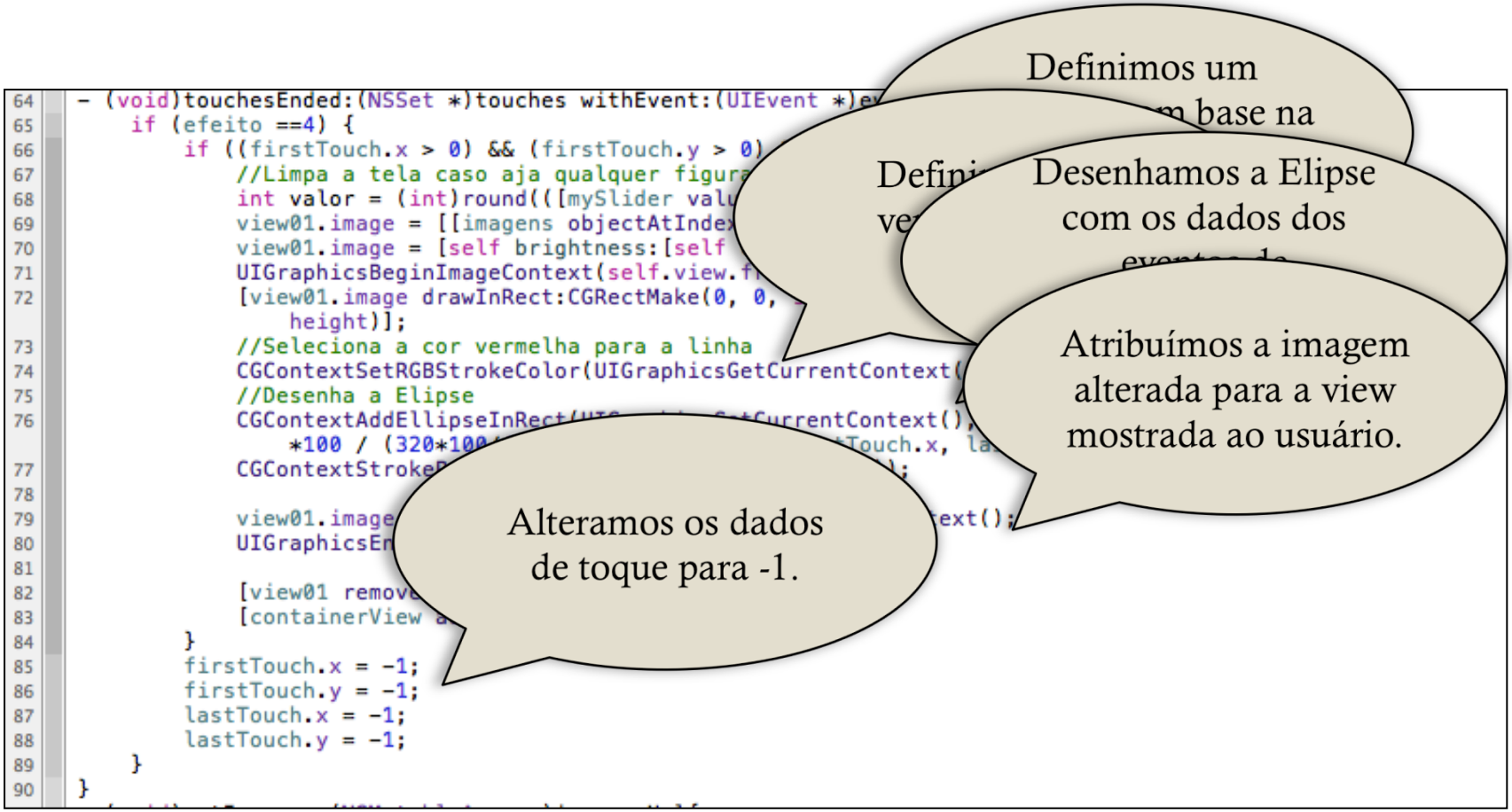
IMPLEMENTAÇÃO - EFEITOS DE BRILHO



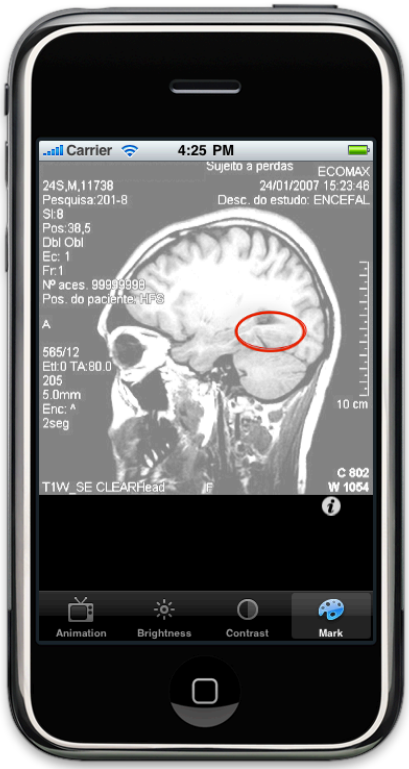

```
85          CGImageRef inImage=
86
87          CGContextRef ctx;
88
89          CFDataRef m_DataRef;
90          m_DataRef = CGDataProviderCopyData(CGImageGet
91          UInt8 * m_PixelBuf = (UInt8 *) CFDataGetBytePtr
92          int length = CFDataGetLength(m_DataRef);
93          CGImageGetBitsPerComponent(inImage), CGImageGetBi
94
95          for (int index = 0; index < length; index += 4)
96          {
97              Byte tempR = m_PixelBuf[index + 1];
98              Byte tempG = m_PixelBuf[index + 2];
99              Byte tempB = m_PixelBuf[index + 3];
100
101             int outputRed = level + tempR;
102             int outputGreen = level + tempG;
103             int outputBlue = level + tempB;
104
105             if (outputRed>255) outputRed=255;
106             if (outputGreen>255) outputGreen=255;
107             if (outputBlue>255) outputBlue=255;
```

VISUALIZAÇÃO DE BRILHO

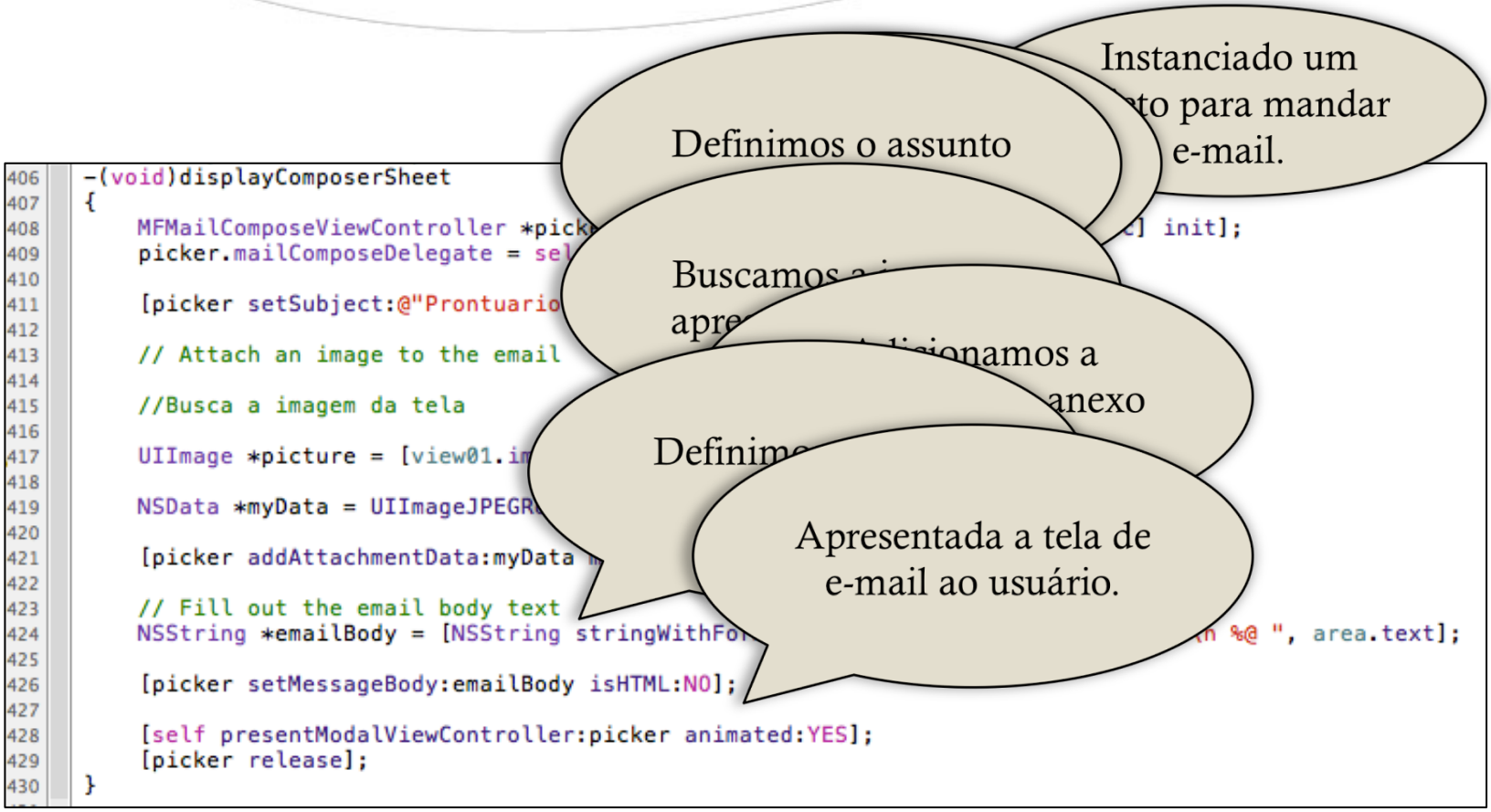
IMPLEMENTAÇÃO – MARCAÇÃO NAS IMAGENS



VISUALIZAÇÃO - MARCAÇÃO NAS IMAGENS



IMPLEMENTAÇÃO – ENVIO E-MAIL




```
406  -(void)displayComposerSheet
407  {
408      MFMailComposeViewController *pick
409      picker.mailComposeDelegate = sel
410
411      [picker setSubject:@"Prontuario
412
413      // Attach an image to the email
414
415      //Busca a imagem da tela
416
417      UIImage *picture = [view01.im
418
419      NSData *myData = UIImageJPEGR
420
421      [picker addAttachmentData:myData
422
423      // Fill out the email body text
424      NSString *emailBody = [NSString stringWithFo
425
426      [picker setMessageBody:emailBody isHTML:NO];
427
428      [self presentModalViewController:picker animated:YES];
429      [picker release];
430  }
```

```
] init];
```

```
h %@ ", area.text];
```

VISUALIZAÇÃO – ENVIO E-MAIL

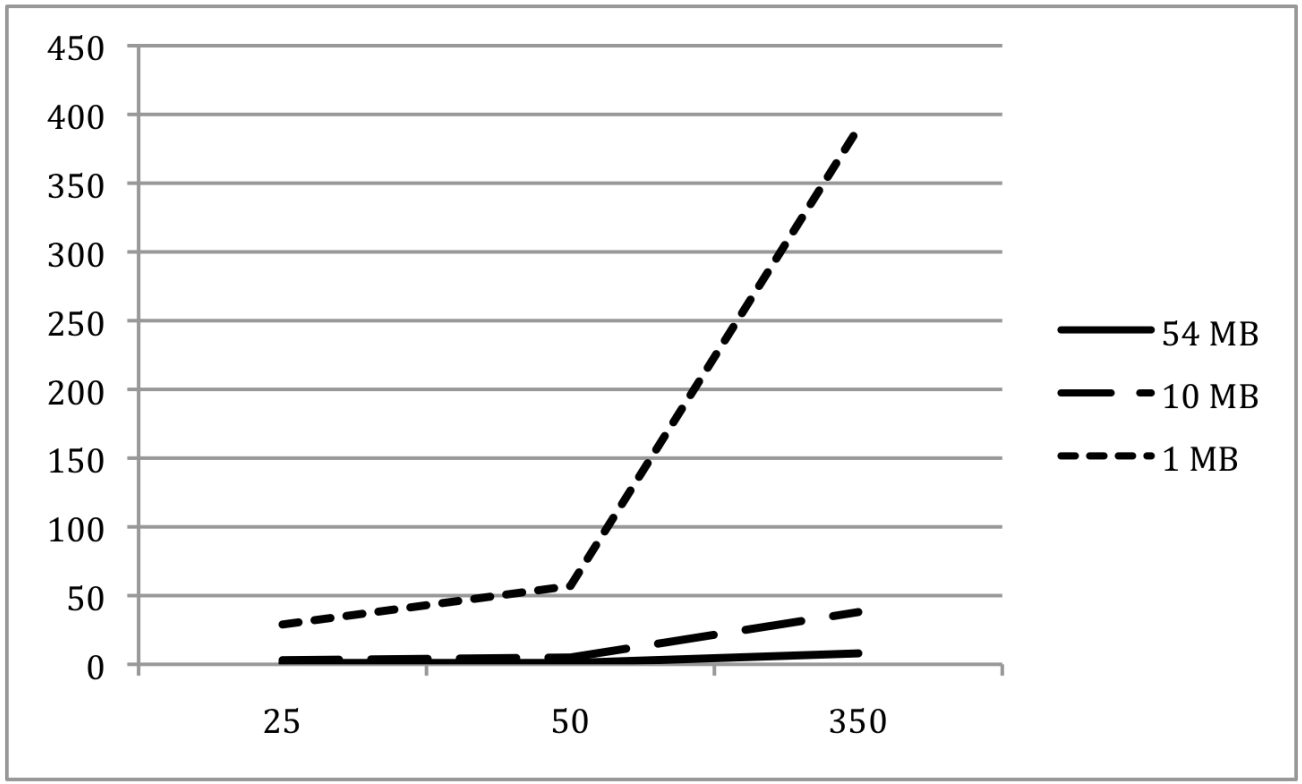
RESULTADOS E DISCUSSÕES

- Tamanho das imagens carregadas
- Relação tempo de carga VS velocidade conexão
- Utilização de memória da aplicação
- Migrar código desktop do Osirix para iPhone

TAMANHO DAS IMAGENS CARREGADAS

Quantidade de Imagens	Velocidade de Conexão	Tamanho Total	Tempo de Carga
25	1 MB	1,3 MB	29 segundos
50		2,6 MB	57 segundos
350		18,5 MB	6 minutos e 30 segundos
25	10 MB	1,3 MB	3 segundos
50		2,6 MB	5 segundos
350		18,5 MB	38 segundos
25	54 MB	1,3 MB	1 segundos
50		2,6 MB	1 segundos
350		18,5 MB	8 segundos

RELAÇÃO TEMPO DE CARGA VS VELOCIDADE CONEXÃO



UTILIZAÇÃO DE MEMÓRIA DA APLICAÇÃO

Quantidade de Imagens	Memória Real	Memória Virtual	Tamanho total das imagens	Memória Real	Memória Virtual
25	4,17 MB	53,53 MB	1,3 MB	9,45 MB	69,57 MB
50			2,6 MB	9,74 MB	69,74 MB
350			18,5 MB	15,68 MB	86,58 MB

CONCLUSÃO

- Disponibilidade de uma forma mais prática e ágil a visualização de imagens radiológicas no formato JPEG
- Explorou a plataforma do desenvolvimento Objective-C para iPhone, com a integração de recursos como: acelerômetro, multitoque e transmissão Web
- Requisitos e objetivos em geral foram alcançados

EXTENSÕES

- Utilizar as imagens no formato DICOM para visualização
- Implementar a persistência local dos arquivos
- Carregar as imagens em lotes
- Autenticação prévia do usuário para o acesso as imagens médicas
- Rotação das imagens usando DICOM

APRESENTAÇÃO PRÁTICA

Obrigado!