



UNIVERSIDADE REGIONAL DE BLUMENAU

VISUALIZAÇÃO DE IMAGENS CAPTURADAS EM UM CIRCUITO FECHADO DE TELEVISÃO (CFTV) NO IPHONE

Diogo Carlassara – Acadêmico

Dalton Solano dos Reis - Orientador



Roteiro

- Introdução
 - Objetivos do trabalho
- Fundamentação teórica
 - CFTV, iPhone, Linguagem de programação, Meios de transmissão de dados, Aplicações CCTV no iPhone
- Desenvolvimento
 - Requisitos principais, especificação, resultados e discussão
- Conclusão
 - Extensões



Introdução

- Segurança
- Monitoramento – CFTV
- Equipamentos
- Visualização através de navegadores
- Mobilidade do dispositivo móvel



Objetivos do trabalho

- Servidor:
 - Captura de imagens
 - Câmera de vídeo (*webcam*)
 - Disponibilizar vídeos para iPhone
- Cliente:
 - Gravação de locais de monitoramento
 - Leitura vídeos nos locais
 - Visualização dos vídeos



UNIVERSIDADE REGIONAL DE BLUMENAU

Fundamentação teórica



CFTV

- Sistema de segurança
- Distribui sinais para pontos específicos
- Equipamentos e estrutura básica
- Custo



Estrutura básica de um sistema de CFTV



iPhone

- Dispositivo móvel: grupo dos *smartphones*
- Funções:
 - Câmera digital
 - Internet sem fio
 - GPS
 - Tela *touch-screen*
- Particularidades:
 - Ausência de teclado
 - *Multi-touch*



Foto de um iPhone



Linguagem de programação

- Linguagem oficial para desenvolvimento: Objective-C
- Base da linguagem:
 - C (velocidade e simplicidade)
 - Smalltalk (capacidade orientação a objetos)
- Características:
 - Simplicidade
 - Herança
 - Polimorfismo
 - Orientação a objetos
 - Não permite criação de classes abstratas



SQLite

- Banco de dados para iPhone
- Permite criação de tabelas
- Manipulação de dados
- Transações – ACID
- Configuração zero
- Suporte até 2 terabytes de dados
- Não é necessária instalação de SGBD
- Banco de dados gravado em arquivo



Meios de transmissão de dados

- Oferece suporte ao fluxo de dados entre dois pontos
 - Redes sem fio
 - Utiliza ondas de rádio
 - Padrão IEEE 802.11b (11 Mbps) ou IEEE 802.11g (54 Mbps)
 - Adaptador de rede traduz dados
 - Transmissão via antena
 - Roteador decodifica
 - Envia a informação utilizando conexão física
 - Redes 3G (gerações x transmissão)
 - 1ª: voz analógica via FDMA
 - 2ª: voz digital via TDMA
 - 3ª : voz e dados digitais em alta velocidade, via WCDMA



Aplicações CCTF no iPhone

- Algumas aplicações CCTF no iPhone:
 - YOICS
 - Cavu Free Video Surveillance
 - NextView Remote Video Camera Surveillance



YOICS

- *Webapp* – Acesso direto via navegador
- Acesso com usuário e senha
- Live View – Visualização em tempo real
- Biblioteca de imagens
- *Mobile explorer*



Telas do Yoics



Cavu Free Video Surveillance

- Visualização em tempo real via internet
- Manipulação de imagens
- Versão gratuita - classe IP



Tela do Cavu



NextView

- Até 16 câmeras em tempo real (cada conta)
- Conexão 3G ou sem fio
- Possibilidade de integração com sistemas de alarmes domésticos



Tela do NextView



Comparação

Software	Yoics	CAVU	NextView
Transmissão em tempo real	✓	✓ *	✓ **
Funciona com rede sem fio	✓	✓	✓
Possui versão gratuita	✓	✓	✓
Visualização de câmeras locais/classe IP	✓	✓	✓
Zoom		✓	✓ **
Câmera pode ser controlada remotamente			✓ **
Visualizada no navegador	✓		
* Apenas na versão paga ** Devem ser adquiridas as câmeras do próprio fabricante			

Comparativo entre os principais softwares de vigilância



UNIVERSIDADE REGIONAL DE BLUMENAU

Desenvolvimento do sistema



Requisitos principais

REQUISITOS FUNCIONAIS	CASO DE USO
RF01: Permitir ao usuário visualizar imagens geradas em computador com uma câmera no iPhone.	UC03, UC04, UC05
RF02: Permitir ao usuário gravar e excluir os locais de acesso do iPhone.	UC01 e UC02
RF03: A aplicação servidora deverá possibilitar ao usuário bater fotos.	UC06

Requisitos funcionais

REQUISITOS NÃO FUNCIONAIS
RNF01: Ser desenvolvido utilizando análise orientada a objetos.
RNF02: As aplicações devem ser desenvolvidas em sistema operacional Mac OS.
RNF03: As aplicações devem ser desenvolvidas utilizando ambiente XCode (para codificação) e Interface Builder (para criação das interfaces).
RNF04: Deverá ser desenvolvido o programa servidor, responsável pela captura das imagens da câmera do computador.
RNF05: Deverá ser desenvolvido o programa cliente, responsável por receber e exibir as imagens geradas pelo servidor.

Requisitos não funcionais



Diagrama de casos de uso

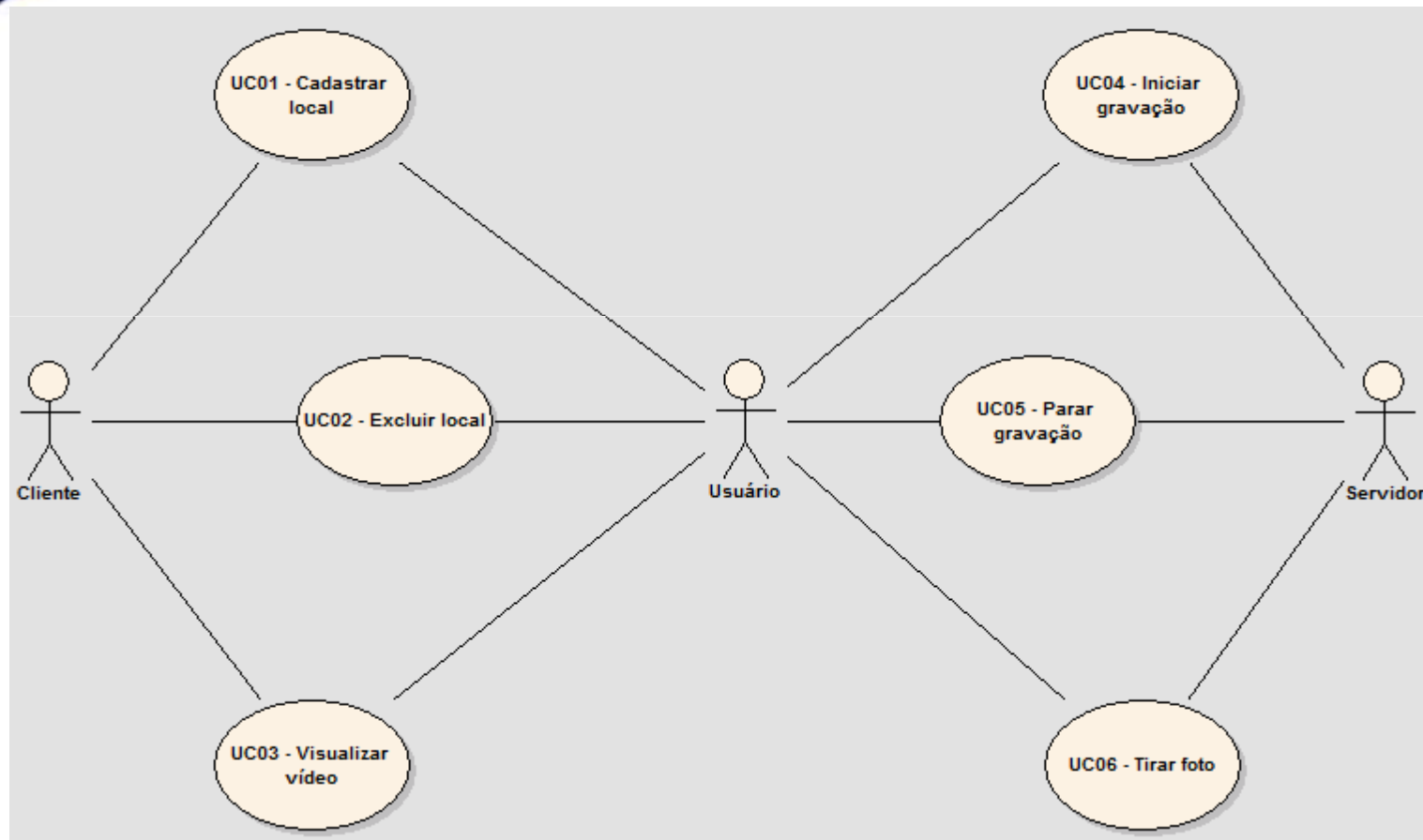


Diagrama de casos de uso

Desenvolvimento da ferramenta → Especificação → Diagrama de casos de uso



Implementação Servidor

class Servidor

MyRecorderController

- btnFoto: NSButton
 - btnStart: NSButton
 - btnStop: NSButton
 - currentImageBuffer: CVMImageBufferRef
 - mCaptureDeviceInput: QTCaptureDeviceInput
 - mCaptureMovieFileOutput: QTCaptureMovieFileOutput
 - mCaptureSession: QTCaptureSession
 - mCaptureView: QTCaptureView
 - mDecompressedVideoOutput: QTCaptureDecompressedVideoOutput
 - mPreviewVideo: QTCaptureVideoPreviewOutput
 - viewImagem: NSImageView
-
- + exibelImagem() : void
 - + startRecording(id) : IBAction
 - + stopRecording(id) : IBAction
 - + tirarFoto(id) : IBAction

Diagrama de classes



Implementação Cliente

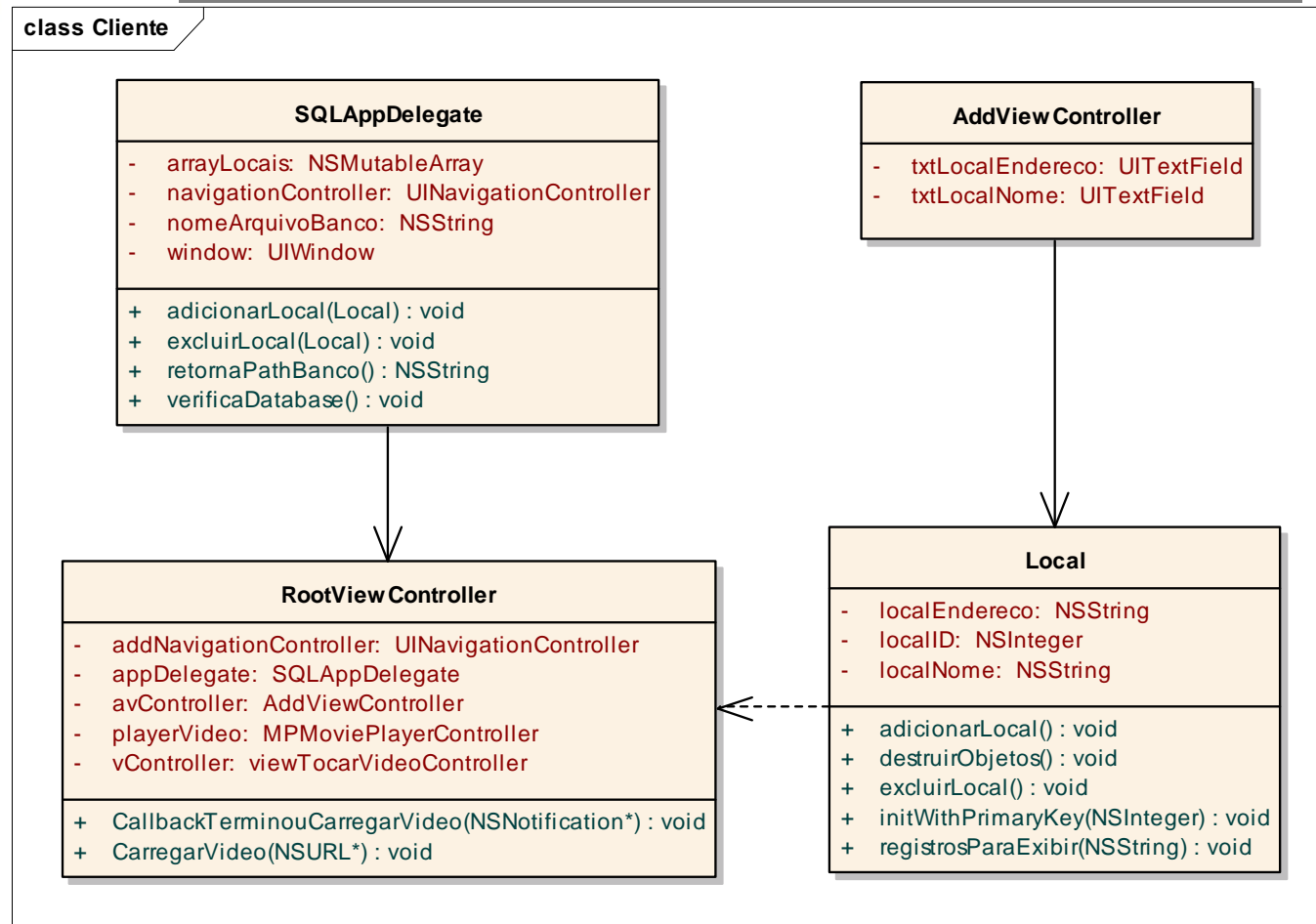
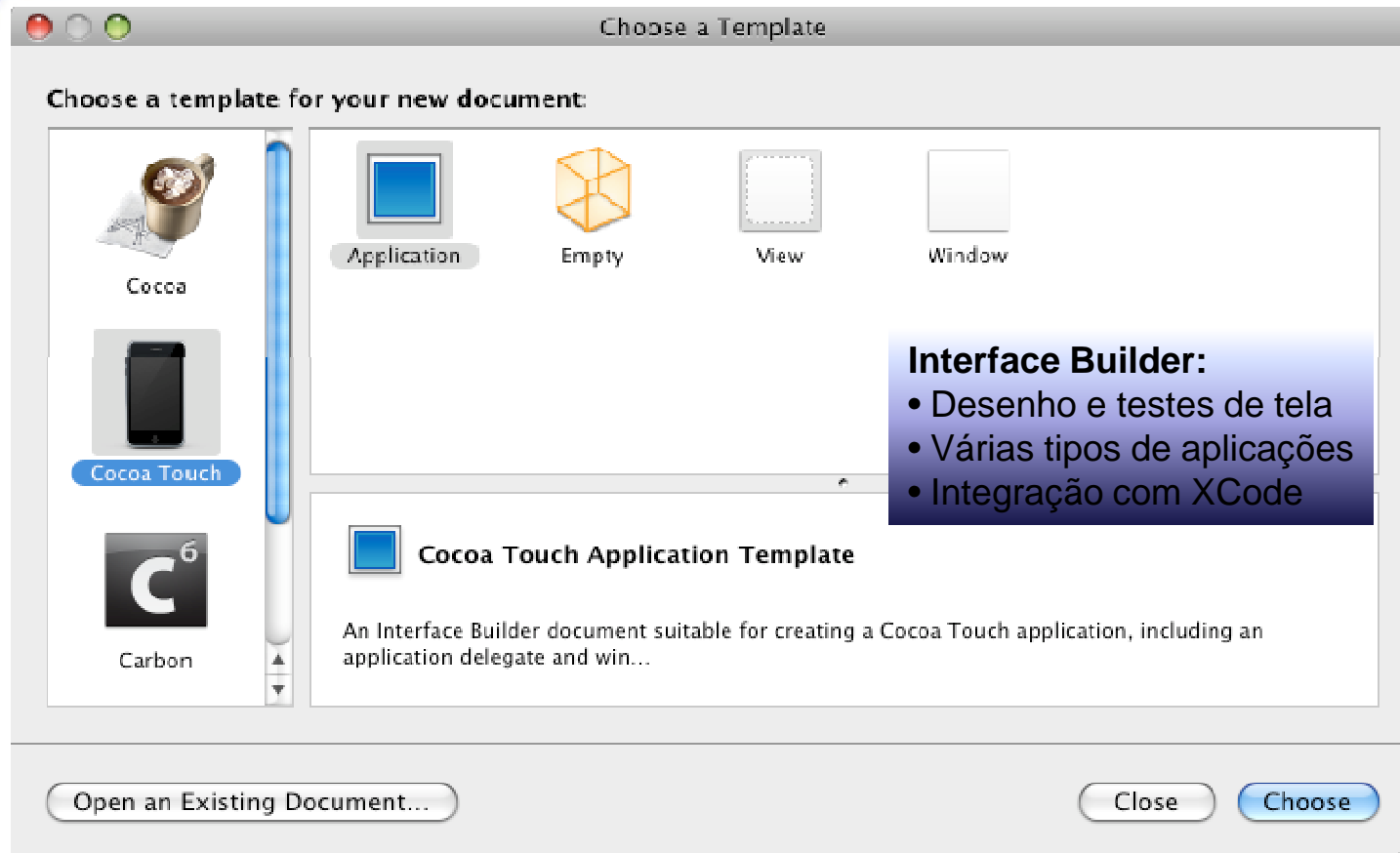


Diagrama de casos de uso



Ferramentas utilizadas



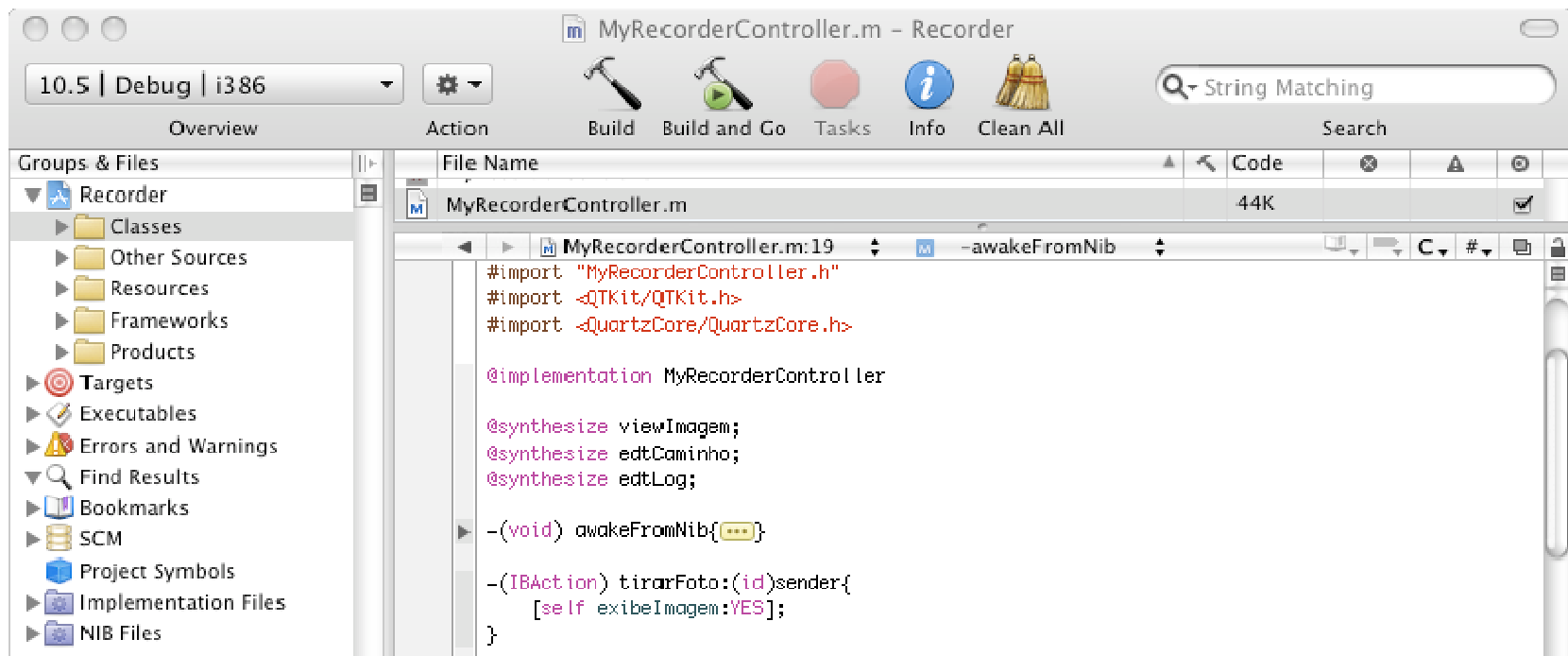
Interface Builder



Ferramentas utilizadas

XCode:

- Ferramenta desenvolvimento Mac OS
- Suporte a várias linguagens, entre elas C, C++, Objective-C



Ambiente XCode

Desenvolvimento da ferramenta → Especificação → Implementação → Ferramentas utilizadas → XCode



Utilizando o sistema - Servidor

The screenshot shows a window titled "Gravador" with a video feed of a classroom. Below the video are controls for recording and snapshots. Red arrows point to numbered annotations:

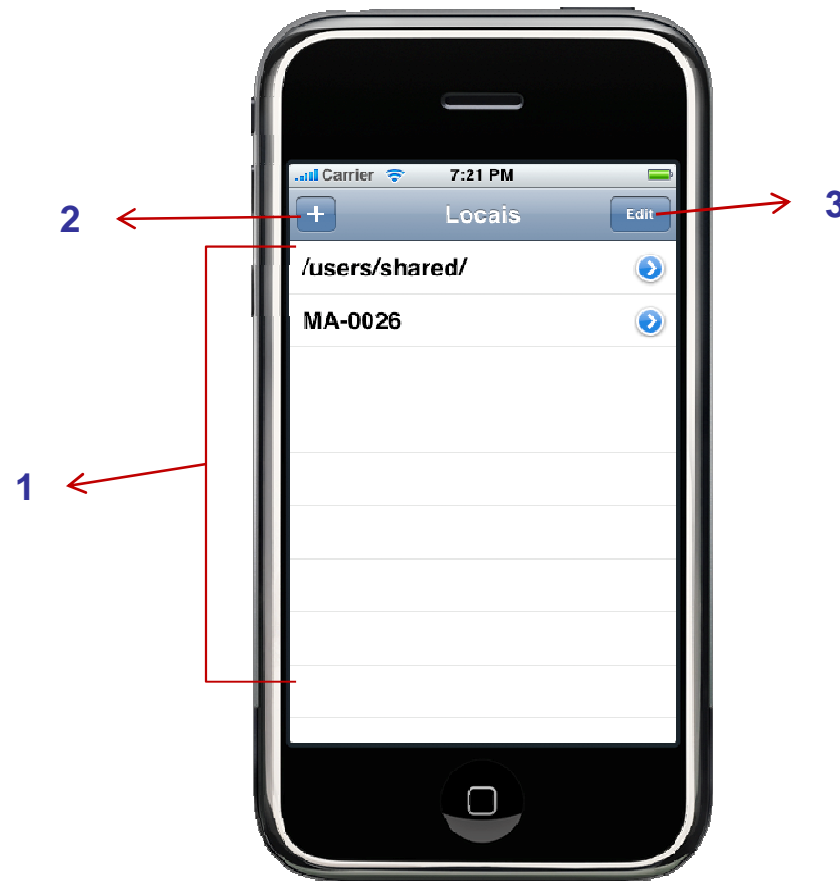
- 1: Points to the video feed area.
- 2: Points to the "iniciar gravação" button.
- 3: Points to the "parar" button.
- 4: Points to the "snapshot" button.
- 5: Points to the "Log:" text area.
- 6: Points to the "Último snapshot:" text area.

Tela da aplicação Servidor

Desenvolvimento da ferramenta → Especificação → Utilizando o sistema - Servidor



Utilizando o sistema - Cliente



Tela da aplicação Cliente



Resultados e discussão

REQUISITOS FUNCIONAIS	CONCLUÍDO
RF01: Permitir ao usuário visualizar imagens geradas em computador com uma câmera no iPhone.	Sim
RF02: Permitir ao usuário gravar e excluir os locais de acesso do iPhone.	Sim
RF03: A aplicação servidora deverá possibilitar ao usuário bater fotos.	Sim

Requisitos concluídos



Servidor

Equipamentos utilizados nos testes:

- a) computador *desktop* modelo iMac;
- b) processador Intel Core 2 Duo 2.4 Gigahertz, com 6 Megabytes de cache L2;
- c) um gigabyte de memória RAM DDR2 800Mhz
- d) placa de vídeo ATI Radeon HD 2400 PCIe x16, com 128 Megabytes de memória RAM;
- e) resolução de vídeo 1680 x 1050, com profundidade de 32 *bits* de cores;
- f) disco rígido de 250 Gigabytes;
- g) câmera *iSight* (embutida), de resolução 640 *pixels* de altura e 480 *pixels* de largura, com possibilidade de captura de 30 frames por segundo (FPS).



Servidor

TEMPO	TAMANHO APROXIMADO
20 segundos	580 Kb
65 segundos	2.1 Mb
90 segundos	2.9 Mb
120 segundos (2 minutos)	3.4 Mb
240 segundos (4 minutos)	9.4 Mb
1320 segundos (22 minutos)	45.2 Mb

Faixas de tempo e tamanho do vídeo

FORMATO	TAMANHO APROXIMADO
“.bmp” – <i>Bitmap</i>	5.2 Mb
“.gif” – <i>Graphics Interchange Format</i>	540 Kb
“.jpg” – <i>Joint Photographic Experts Group</i>	252 Kb
“.png” – <i>Portable Network Graphics</i>	2.2 Mb

Comparação entre os formatos de imagens



Cliente

- Ler vídeos disponibilizados pelo Servidor
 - Emulador iPhone versão 2.2.1
 - Xcode versão 3.1.3
 - Frequência processador iPhone: 620 mhz
-
- Banco de dados com 5 registros cadastrados: 5 Kb
 - Total da aplicação: 80 Kb



Conclusão

- Criado um receptor e visualizador de *streaming* de vídeo
- Servidor, instalado no computador, armazena o vídeo da *webcam* em arquivos
- Cliente, instalado no iPhone, efetua a leitura dos vídeos
- Aplicações com alta performance
- Ambiente de testes para outras aplicações



Extensões

TAREFA	COMPLEXIDADE
Permitir à aplicação Cliente fotografar os locais de monitoramento, armazenando as imagens no iPhone.	4
Permitir que a aplicação Servidor suporte várias câmeras ao mesmo tempo e em tempo real.	6
Criar recursos de integração entre a aplicação Servidor com o sistema de alarme doméstico, detectando a abertura e/ou fechamento de portas e janelas, enviando alertas à aplicação Cliente.	8
Criar mecanismo de troca de mensagens entre a aplicação Cliente e Servidor, a fim de implementar maiores controles sobre as mesmas.	6
Permitir à aplicação Servidor enviar <i>streaming</i> em tempo real para a aplicação Cliente.	7
Permitir que as aplicações Servidor e Cliente ofereçam recursos de gravação de vídeo com áudio.	4

Lista de extensões



UNIVERSIDADE REGIONAL DE BLUMENAU

Apresentação prática



UNIVERSIDADE REGIONAL DE BLUMENAU

Obrigado!

“Por mais que o preguiçoso deseje alguma coisa, ele não conseguirá, mas a pessoa esforçada consegue o seu desejo.

Provérbios 13:4