

**UNIVERSIDADE FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS DE INFORMAÇÃO**

APLICATIVO PARA ERGONOMIA DE APRESENTAÇÕES

**CURITIBA
2009**

**FÁBIO SILVA KRIEGER
JOÃO MANOEL DANNEMANN
RACHID DEQUÊCH**

APLICATIVO PARA ERGONOMIA DE APRESENTAÇÕES

Trabalho de Conclusão do Curso de Sistemas de Informação, da Universidade Federal do Paraná, como requisito final à obtenção do diploma de tecnólogo em sistemas de informação.

Orientador: Roberto Tadeu Raittz

**CURITIBA
2009**

AGRADECIMENTOS

Agradecemos aos professores que nos ajudaram direta ou indiretamente para tornar o aplicativo realidade. Em especial ao nosso professor orientador Roberto Tadeu Raitz, ao professor Dieval Guizelini e também ao professor Lucas Ferrari de Oliveira. Agradecemos também pelas dicas dadas por nossos companheiros de curso e pelos alunos do mestrado em bioinformática de 2009.

RESUMO

O aplicativo desenvolvido tem como objetivo auxiliar pessoas que utilizam projetores para realizar suas apresentações. Com a emissão de luz de cor preta no corpo e nos olhos do apresentador, o aplicativo proporciona maior conforto para o mesmo e também o deixa livre de ser prejudicado a longo prazo, como problemas de visão, dores de cabeça, entre outros, pois a referida luz representa ausência de cor.

Para a realização do projeto foram utilizadas técnicas de comparação de imagens por diferença, algoritmos genéticos, manipulação de frames de webcam. O aplicativo ainda conta com um apresentador de slides integrado, capaz de exibir na tela em modo de apresentação, imagens .gif e .jpg. Estas imagens são importadas de um diretório previamente definido.

Palavras-chave:

Algoritmo Genético, Subtração de Imagens, Apresentação

ABSTRACT

The developed application is used to help people who use projectors to make their presentations. With the black light issue in the body and in the eyes of the presenter, the application provides him more comfort , and also leaves him free from being harmed in the long run, such as vision problems, headaches, among others, for the black light means absence of color.

To achieve the project objectives, were used: comparison of images by difference techniques, genetic algorithms and handling of webcam frames. The application also has an integrated slide presenter, able to display on screen in slideshow mode, images .gif and jpg. These images are imported from a previously defined directory.

Keywords:

Genetic algorithm, Image Subtraction, Presentation

LISTA DE FIGURAS

Figura 1 – Imagem normal 1	20
Figura 2 – Imagem resultado 1	20
Figura 3 – Imagem normal 2	21
Figura 4 – Imagem resultado 2	21

LISTA DE DIAGRAMAS

Diagrama 1 – Fluxograma do funcionamento base do programa	27
Diagrama 2 – Caso de uso – Iniciar apresentação	29
Diagrama 3 – Caso de uso – Avançar slide	31
Diagrama 4 – Caso de uso – Voltar slide	33
Diagrama 5 – Caso de uso – Sincronizar projeção	35
Diagrama 6 – Diagrama de sequência – Iniciar apresentação	37
Diagrama 7 – Diagrama de sequência – Avançar slide	38
Diagrama 8 – Diagrama de sequência – Voltar slide	39
Diagrama 9 – Diagrama de sequência – Sincronizar slide	40
Diagrama 10 – Diagrama de Classes	41

LISTA DE TABELAS

Tabela 1 – Tabela de Necessidades.....	13
Tabela 2 – Recursos Materiais.....	24
Tabela 3 – Problemas encontrados e contornados.....	25

SUMÁRIO

1. PROBLEMATIZAÇÃO	12
2. OBJETIVOS	14
2.1 OBJETIVO GERAL	14
2.2 OBJETIVOS ESPECÍFICOS	14
3. FUNDAMENTOS TEÓRICOS	15
3.1 TECNOLOGIAS UTILIZADAS.....	15
3.1.1 Java	15
3.1.2 UML (Unified Modeling Language).....	15
3.2 FERRAMENTAS UTILIZADAS	16
3.2.1 Integrated Development Environment (IDE)	16
3.2.2 JUDE (Java and UML Developer Environment).....	17
3.3 TÉCNICAS UTILIZADAS	17
3.3.1 Captura de Frame	17
3.3.2 Algoritmo Genético.....	17
3.3.3 Comparação de Imagens.....	19
3.3.4 Subtração de imagens pixel a pixel.....	19
3.3.5 Redimensionamento de Imagem	22
3.3.6 Controle da Câmera e Captura de Imagem	22
3.3.7 Procura de Coordenadas de Projeção	22
3.3.8 Apresentador de Slides.....	22
4. RECURSOS	24
4.1 HUMANOS.....	24
4.2 MATERIAIS.....	24
5. PROBLEMAS ENCONTRADOS E CONTORNADOS	25
5.1 DIFICULDADES DE INFRA-ESTRUTURA	25
5.2 DIFICULDADES TÉCNICAS.....	25
5.3 DIFICULDADES PESSOAIS.....	25
6. O APLICATIVO PARA ERGONOMIA DE APRESENTAÇÕES	27
6.1 ARQUITETURA DO SISTEMA	28
6.2 CASOS DE USO.....	29
6.3 DIAGRAMAS DE CASO DE USO	29
6.3.1 UC001 – Iniciar Apresentação	29
6.3.2 UC002 – Avançar Slide.....	31
6.3.3 UC003 – Voltar Slide.....	33
6.3.4 UC004 – Sincronizar Projeção	35
6.4 REGRAS DE NEGÓCIO	36
6.5 DIAGRAMAS DE SEQUÊNCIA.....	37
6.5.1 DS001 – Iniciar Apresentação	37
6.5.2 DS002 – Avançar Slide	38
6.5.3 DS003 Voltar Slide.....	39
6.5.4 DS004 Sincronizar Slide	40
6.6 DIAGRAMA DE CLASSES.....	41
7 CONCLUSÃO E TRABALHOS FUTUROS	42
8 ANEXOS	43
8.5 GUIA DE INSTALAÇÃO.....	43
8.5.2 Requisitos Mínimos.....	43
8.5.3 Passos para Instalação.....	43

9 REFERÊNCIAS	44
10 GLOSSÁRIO	45

INTRODUÇÃO

A utilização de ferramentas para auxiliar nas apresentações dos mais variados tipos vem evoluindo muito em nosso meio, tendo seu início com cartazes, passando por retroprojetores e hoje em dia sendo a grande maioria baseada em datashow. Essa evolução tende a facilitar a vida do ser humano, pois os novos meios de realizarmos nossas apresentações são mais práticos e modernos. Podemos afirmar isto levando em consideração a diferença de uma apresentação feita com um antigo retroprojetor – que necessitava uma impressão em lâmina transparente do que se desejava apresentar, folha por folha – para uma apresentação com um moderno datashow, que é conectado em um computador e pode reproduzir inúmeros arquivos e tipos de apresentação do mesmo. Porém, aparentemente essa evolução estacionou, pois um grande problema de utilizar este aparelho de multimídia não foi sanado. Trata-se da emissão de luz na pessoa que está realizando a apresentação, principalmente na região dos olhos, o que pode trazer além de malefícios para a saúde deste sensível órgão do nosso corpo uma diminuição da qualidade da apresentação.

1. PROBLEMATIZAÇÃO

O sistema desenvolvido é um aplicativo para ergonomia de apresentações. Trata-se de uma ferramenta que diminui a irritação dos olhos pela incidência de luz de um datashow. Durante uma apresentação de slides, o apresentador se encontra em diversos momentos exposto ao feixe de luz emitido pelo projetor, o que pode prejudicar seriamente sua visão tanto durante a apresentação como a longo prazo, além de expô-lo a uma situação vexatória, pois tanto o texto exposto como as figuras serão projetados em seu próprio corpo.

Uma boa apresentação aliada a um bom apresentador é garantia de total entendimento e convencimento das pessoas que estão assistindo. Porém, quando um desses princípios é prejudicado, o resultado pode ser desastroso. Sendo assim, temos um problema: há a necessidade do total bem-estar do apresentador, não podendo sofrer interferências principalmente do datashow, que é um aliado para o seu sucesso. Além de ofuscar a vista, a projeção das imagens do datashow no corpo do apresentador pode desviar a atenção das pessoas que estão assistindo a apresentação. As principais necessidades de utilização do aplicativo estão listadas na tabela seguinte:

Necessidade	Prioridade	Preocupações	Solução Atual	Solução Proposta
Evitar ofuscamento	Média	Ofuscamento pode prejudicar a apresentação, pois a visão do apresentador para a <u>sala</u> e para a apresentação é indispensável.	Não há.	Com nosso aplicativo, a luz não incidirá sobre os olhos do apresentador, evitando assim o ofuscamento.
Evitar problemas de saúde	Alta	Devem ser evitados por questões de segurança no trabalho.	Não há.	Sem a incidência da luz forte diretamente nos olhos do apresentador, diminuem os riscos de Astenopia, lesões na retina, entre outros
Evitar luz no corpo e desvio de atenção	Alta	Caso ocorram estas hipóteses, a apresentação será comprometida.	Não há.	Sem a incidência de elementos da apresentação no apresentador, vexames e desvio de atenção não ocorrerão.

Tabela 1 – Tabela de Necessidades

2. OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral do projeto é a criação de um aplicativo de auxílio ergonômico a apresentações realizadas utilizando um datashow.

2.2 OBJETIVOS ESPECÍFICOS

- Dotar o aplicativo da habilidade de projeção de luz de cor preta nos olhos e no corpo do apresentador, evitando ofuscamento nos olhos e desvios de atenção resultantes da projeção de imagens no corpo do mesmo;
- Estender a utilização do software à profissionais de variadas áreas, pois todos em algum momento tiveram ou terão que fazer uma apresentação para o público, e podemos considerar que uma taxa de 90% das pessoas utilizarão um datashow para fazê-la.

3. FUNDAMENTOS TEÓRICOS

3.1 TECNOLOGIAS UTILIZADAS

3.1.1 Java

A linguagem de programação escolhida para o desenvolvimento do aplicativo foi Java. Baseada no C++, foi especificamente projetado para ser menor e mais simples, não tem ponteiros, mas sim tipo de referência, não é possível escrever subprogramas independentes, possui somente herança simples e interfaces, e coleta de lixo implícita, é muito utilizada para programação para web [3]. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um "bytecode" que é executado por uma máquina virtual. Por ser uma linguagem orientada a objetos é muito popular e compatível com vários sistemas operacionais encontrados hoje no mercado.

3.1.2 UML (Unified Modeling Language)

Devido à rápida percepção dos benefícios da OO (Orientação a Objetos), verificou-se o surgimento de um grande número de propostas de análise e projeto orientados a objeto, tanto dos meios acadêmicos quanto empresariais. As primeiras propostas de notações para a Análise OO surgiram no início da década de 90 e logo após houve uma miscelânea de notações, gerando um excesso de propostas para a diagramação e construção de sistemas com técnicas OO. O que dificultou a adoção das técnicas OO foi a falta de um acordo na escolha do método a seguir. Entre as propostas de Análise OO mais robustas, destacam-se os seguintes autores:

- Peter Coad / Edward Yourdon com Análise Baseada em Objetos
- Jacobson com OOSE (*Object-Oriented Software Engineering*)
- Rumbaugh com o método OMT (*Object Modeling Technique*)
- Grady Booch

Diante de várias técnicas fez com que a OMG [5] (Object Management Group - Consórcio internacional de empresas que define e ratifica padrões na área da

orientação a objetos) aprovasse a UML em 1997. A UML segundo o autor Bezerra [6] é uma mistura de sintaxe gráfica preexistentes, com alguns elementos removidos e outros elementos adicionados com o objetivo de torná-la mais expressiva. A UML é destinada para visualizar, especificar, construir e documentar sistemas complexos.

Segundo Booch [7], a UML é adequada para a modelagem de sistemas, cuja abrangência poderá incluir desde sistemas de informação corporativos a serem distribuídos em aplicações baseadas em Web, até sistemas complexos embutidos em tempo real.

Conforme Booch [7], diagrama é a representação gráfica de um conjunto de elementos, geralmente representados como gráficos de itens e relacionamentos. É através dos diagramas que se pode visualizar o sistema. É difícil um único tipo de diagrama representar o ciclo de vida de um projeto, por isso, a UML disponibilizou vários tipos de diagramas para mostrar a estrutura e comportamento de um sistema sob várias perspectivas.

. Com a UML podemos documentar nossas idéias quando elas já estiverem bem consolidadas para que novos integrantes e novos colaboradores possam acelerar sua compreensão dos sistemas desenvolvidos pelo grupo.

3.2 FERRAMENTAS UTILIZADAS

3.2.1 Integrated Development Environment (IDE)

A IDE utilizada foi o Eclipse GALILEO Java EE IDE for Web Developers, uma IDE freeware. A plataforma Eclipse é uma proposta de consórcio de empresas que apóiam o uso de uma arquitetura aberta para a criação de ambientes integrados de desenvolvimento (IDEs), onde a indústria de software possa desenvolver diversos programas, aplicativos e ferramentas, de forma otimizada e padronizada, baseando-se nas iniciativas de software livre. O Eclipse oferece uma estrutura flexível, pois utiliza linguagem Java (com suporte total para Java) e vem com exemplos de construção. Isso torna mais fácil a criação, integração e utilização das ferramentas, economizando tempo e dinheiro. O eclipse tem sido desdobrado em uma grande escala de desenvolvimento de estações de trabalho como HP-UX, Solaris, AIX e Linux [8]. Disponível em <http://www.eclipse.org/downloads/>

3.2.2 JUDE (Java and UML Developer Environment)

A ferramenta para desenvolvimento dos diagramas UML escolhida foi o JUDE, versão Community Release 5.2. JUDE ou Java and UML Developer Environment é uma das ferramentas freeware para UML com mais funções disponíveis atualmente. Com características que não são encontradas nas outras ferramentas free, como adicionar métodos no diagrama de sequência e a alteração refletir-se no diagrama de classes. O JUDE não possui um apelo gráfico tão bom quanto o Poseidon, ou tantas funcionalidades como Rational Rose, Together e Magic Draw. Seu desempenho impressiona, principalmente tratando-se de uma ferramenta 100% Java/Swing, desmistificando que Swing é lento. Disponível em <http://jude.change-vision.com/jude-web/download/index.html>

3.3 TÉCNICAS UTILIZADAS

3.3.1 Captura de Frame

Um vídeo é composto por diversos frames. Podemos tomar como exemplo um filme de cinema, que é nada mais nada menos que um “rolo de frames” exibidos a uma taxa entre 20 e 30 frames por segundo. A olho nu a junção destes frames sequencialmente passa a impressão de continuidade. Caso a taxa de frames diminua, o cérebro irá emitir a sensação de que perdemos uma cena do filme ou de um pulo de imagem. No aplicativo desenvolvido a captura de frames é constante, utilizada em todos os momentos de comparação de imagens.

3.3.2 Algoritmo Genético

Os algoritmos genéticos são uma família de modelos computacionais inspirados na evolução, que incorporam uma solução potencial para um problema específico numa estrutura semelhante à de um cromossomo e aplicam operadores de seleção e "cross-over" a essas estruturas de forma a preservar informações críticas relativas à solução do problema. Normalmente os AG's (algoritmos genéticos) são vistos como “otimizadores” de funções, embora a quantidade de problemas para o qual os AG's se aplicam seja bastante abrangente. [9]

JGAP é uma API (Application Programming Interface - Interface de Programação de Aplicativos))Java que provê mecanismos básicos para a programação de sistemas evolucionários (algoritmos genéticos) e foi utilizada para o desenvolvimento do aplicativo. Ela foi projetada para exigir o mínimo de esforço na sua utilização, além de ser bastante flexível através da implementação de plugins.

Algoritmos genéticos são implementados como uma simulação de computador em que uma população de representações abstratas de solução é selecionada em busca de soluções melhores. A evolução geralmente se inicia a partir de um conjunto de soluções criado aleatoriamente e é realizada por meio de gerações. A cada geração, a adaptação de cada solução na população é avaliada, alguns indivíduos são selecionados para a próxima geração, e recombinados ou mutados para formar uma nova população. A nova população então é utilizada como entrada para a próxima iteração do algoritmo.

Para configurar o algoritmo genético, deve-se definir vários parâmetros na seguinte ordem:

1. Decidir o número de genes que cada cromossomo terá;
2. Decidir os parâmetros que serão utilizados nos genes, assim como seus valores máximos e mínimos;
3. Determinar a população mínima e o número máximo de evoluções;
4. Escrever a função de avaliação dos cromossomos.

Para determinar a área de projeção na imagem da webcam foram utilizados os seguintes parâmetros:

- 1 - É criada uma população com 80 cromossomos (compostos por 4 genes que correspondem as coordenadas(x,y) de início e final da projeção). Estes genes não podem ser maiores que a área máxima da projeção
2. Os valores mínimos estipulados foram: 0 para X e Y iniciais e finais. Os valores máximos estipulados foram: 352 para X final e inicial (a largura da imagem é de 352 pixels) e 288 para altura (a altura da imagem é de 288 pixels).

3. A população mínima foi configurada para 80 cromossomos e o número de evoluções para 40.

4. Estes cromossomos são evoluídos (cross-over) de acordo com a seguinte fitness:

- Adiciona-se 1 ponto para cada pixel claro na área do cromossomo
- Subtrai-se 1 ponto para cada pixel escuro na área do cromossomo
- Retorna a pontuação daquele cromossomo

Após as 40 evoluções, são extraídos do cromossomo mais adaptado os melhores valores de X e Y iniciais e finais. Com esses valores é então determinada a área do frame onde está a projeção.

3.3.3 Comparação de Imagens

A comparação de imagens foi necessária para encontrar a imagem ideal que deve ser projetada para que haja a ausência de luz. Para isto, temos como base uma imagem do início da apresentação, o primeiro frame captado pela webcam, a qual comparamos com a imagem atual recebida frame a frame, e projetamos a imagem resultado, que é uma imagem com preto sobre o apresentador, a qual representa a diferença entre as duas imagens comparadas. Para chegamos na imagem resultado, técnica de subtração de imagens foi utilizada.

3.3.4 Subtração de imagens pixel a pixel

A subtração em duas imagens é, primeiramente, uma maneira de descobrir diferenças entre elas [1]. Em sistemas de processamento de imagens, o movimento é originado a partir de um deslocamento relativo entre o sistema sensor e a cena que está sendo observada. Uma das abordagens mais simples para a detecção de mudanças entre duas imagens é a comparação das duas imagens pixel a pixel [2] estabelecendo um threshold, onde a partir deste ponto o valor será considerado uma diferença. Esta comparação resultará em uma terceira imagem que será adotada como a imagem da diferença, onde todas as características que não foram alteradas são excluídas, enquanto que as que mudaram permanecem [1]. Como este filtro indica o que há de mudança nas imagens da diferença, ele geralmente é

utilizado em mecanismos de detecção de movimento. Primeiramente, tentamos comparar as imagens utilizando um filtro para transformá-las em escala de cinza e depois realizando a subtração. O resultado foi o seguinte:



Figura 1 – Imagem normal 1

Após transformar em escala de cinza e realizar a comparação com o primeiro frame da câmera (sem interferência), obtivemos o seguinte resultado:

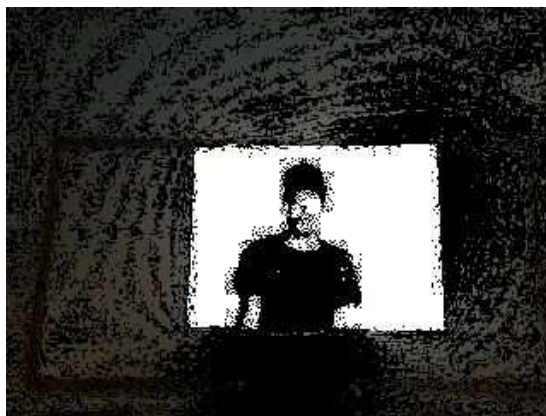


Figura 2 – Imagem resultado 1

Porém, este não era o resultado esperado. Sendo assim, realizamos outros testes com a imagem, até que a solução foi encontrada realizando apenas a subtração das imagens, só que agora sem filtrá-las para escala de cinza. O resultado foi o seguinte:



Figura 3 – Imagem normal 2

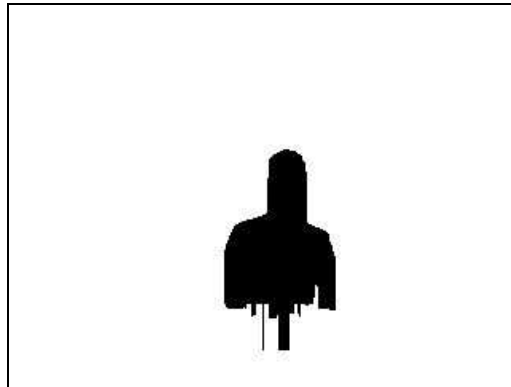
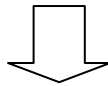


Figura 4 – Imagem resultado 2

Enfim, chegamos ao resultado que procurávamos, pois a imagem a ser usada nas projeções é a que está dentro da tela de projeção, e a transformação da Figura 3 na Figura 4 tornou-se base para as transformações utilizadas no aplicativo.

Para determinar se existe interferência na imagem, a diferença entre um pixel de uma imagem para a outra deve ser maior que um número determinado previamente no código do programa. Este número foi obtido após vários testes, até chegarmos a um resultado satisfatório.

3.3.5 Redimensionamento de Imagem

Como foi citado acima, a imagem da câmera utilizada na comparação não vai estar no mesmo tamanho do slide que estará sendo projetado. Sendo assim, precisaremos processar os resultados obtidos através da comparação, utilizando uma escala padrão entre o tamanho do slide e o tamanho do recorte da projeção do frame da webcam. A dimensão escolhida como padrão foi 800x600 dpi (dots per inch), que é a resolução ideal sugerida pelo próprio aparelho.

3.3.6 Controle da Câmera e Captura de Imagem

Para conseguir fazer com que nosso sistema identificasse a imagem do slide e também fosse capaz de identificar interferências na mesma, utilizamos um dispositivo de captura de vídeo (webcam). Este dispositivo envia ao programa, a cada troca de slide, uma imagem base sem interferência do slide, e também a cada frame processado, envia uma imagem com o frame atual, que pode conter interferências do apresentador.

3.3.7 Procura de Coordenadas de Projeção

Para agilizar o processamento do aplicativo, precisávamos trabalhar com imagens de dimensões reduzidas. Para isso, utilizamos um algoritmo genético que irá, no início da execução do aplicativo ou quando receber o comando do apresentador, enviar uma imagem com um slide um branco e retornar as coordenadas da área de projeção na imagem da câmera. Com estas coordenadas podemos usar apenas as imagens da área de projeção na comparação.

3.3.8 Apresentador de Slides

Para aumentar a comodidade do apresentador e facilitar na pintura da interferência, incorporamos ao nosso aplicativo um apresentador de slides. Neste primeiro momento o apresentador funciona acessando um diretório pré-estabelecido no disco rígido, acessando imagens no formato .JPG e .GIF da pasta e exibindo-as na tela em modo *Full Screen*. Este apresentador contém as funções de avançar o

slide (botão esquerdo do mouse e seta direita do teclado), voltar o slide (botão direito do mouse e seta direita do teclado) e sincronizar a área de projeção com a webcam (tecla F5).

4. RECURSOS

4.1 HUMANOS

A equipe que desenvolveu o projeto é composta por 3 (três) membros do curso de Tecnologia em Sistemas de Informação:

- Fábio Silva Krieger
- João Manoel Dannemann
- Rachid Dequêch

Porém, os recursos humanos se estendem a outras pessoas, como o professor orientador Roberto Tadeu Raittz, que foi essencial e motivador para o direcionamento e alinhamento da equipe, professor Dieval Guizelini, professor Lucas Ferrari de Oliveira, entre outros.

4.2 MATERIAIS

Os recursos materiais utilizados no projeto são computadores portáteis (notebooks), computadores pessoais (desktops), retroprojetores, webcams e uma TV. A seguir é apresentada tabela com os recursos materiais:

Recurso	Modelo / Configuração
Notebook 1	HP dv5, Core 2 Duo 2.0 GHz, 4,00 GB RAM, Windows Vista
Notebook 2	HP dv6, Core 2 Duo 2.0 GHz, 4,00 GB RAM, Windows Vista
Desktop	Core 2 Duo 2.0 GHz, 4,00 GB RAM, Windows 7
Projektor 1	Projektor Benq MP511+ Multimídia - 2100 Lumens - DLP
Projektor 2	Hitachi CP-S335
Webcam 1	Integrada ao notebook HP dv5
Webcam 2	LG WebPro ²
Webcam 3	Logitech
TV	Samsung A550 32"

Tabela 2 – Recursos Materiais

5. PROBLEMAS ENCONTRADOS E CONTORNADOS

Infra-Estrutura	Técnicos	Pessoais
Datashow	Ruídos na imagem	Irritabilidade
Espaço físico	Sensibilidade a luz do ambiente	Falta de tempo
	Rapidez no processamento da comparação	

Tabela 3 – Problemas encontrados e contornados

5.1 DIFICULDADES DE INFRA-ESTRUTURA

-DataShow

-Espaço físico

O grande problema de infra-estrutura encontrado foi o de um recurso fundamental para o desenvolvimento do aplicativo: o datashow. Porém, nas dependências da faculdade dispusemos do aparelho para realizarmos nossos testes e a implementação de nosso aplicativo.

5.2 DIFICULDADES TÉCNICAS

- Ruídos na imagem

- Rastros na imagem

- Sensibilidade a iluminação do ambiente

- Velocidade/Agilidade no processamento da comparação

5.3 DIFICULDADES PESSOAIS

- Irritabilidade

- Falta de Tempo

Dados os testes exaustivos aliados à falta de tempo, a irritabilidade apareceu em certos momentos no grupo, porém foi rapidamente contornada. O tempo tornou-se curto, pois além da decisão um pouco tardia do tema do projeto, que durou várias reuniões com o grupo e com o professor orientador, trata-se de um projeto de

complexidade muito alta. Sendo assim, em muitos momentos os problemas encontrados demoraram para ser contornados, pois a experiência do grupo era ínfima em relação à complexidade. A ajuda do corpo docente do curso de Tecnologia em Análise de Sistemas de Informação foi essencial para contornar os referidos problemas.

6. O APLICATIVO PARA ERGONOMIA DE APRESENTAÇÕES

Com a união dos recursos técnicos, tecnológicos e humanos, o aplicativo foi criado. No fluxograma abaixo podemos ver o funcionamento base do programa:

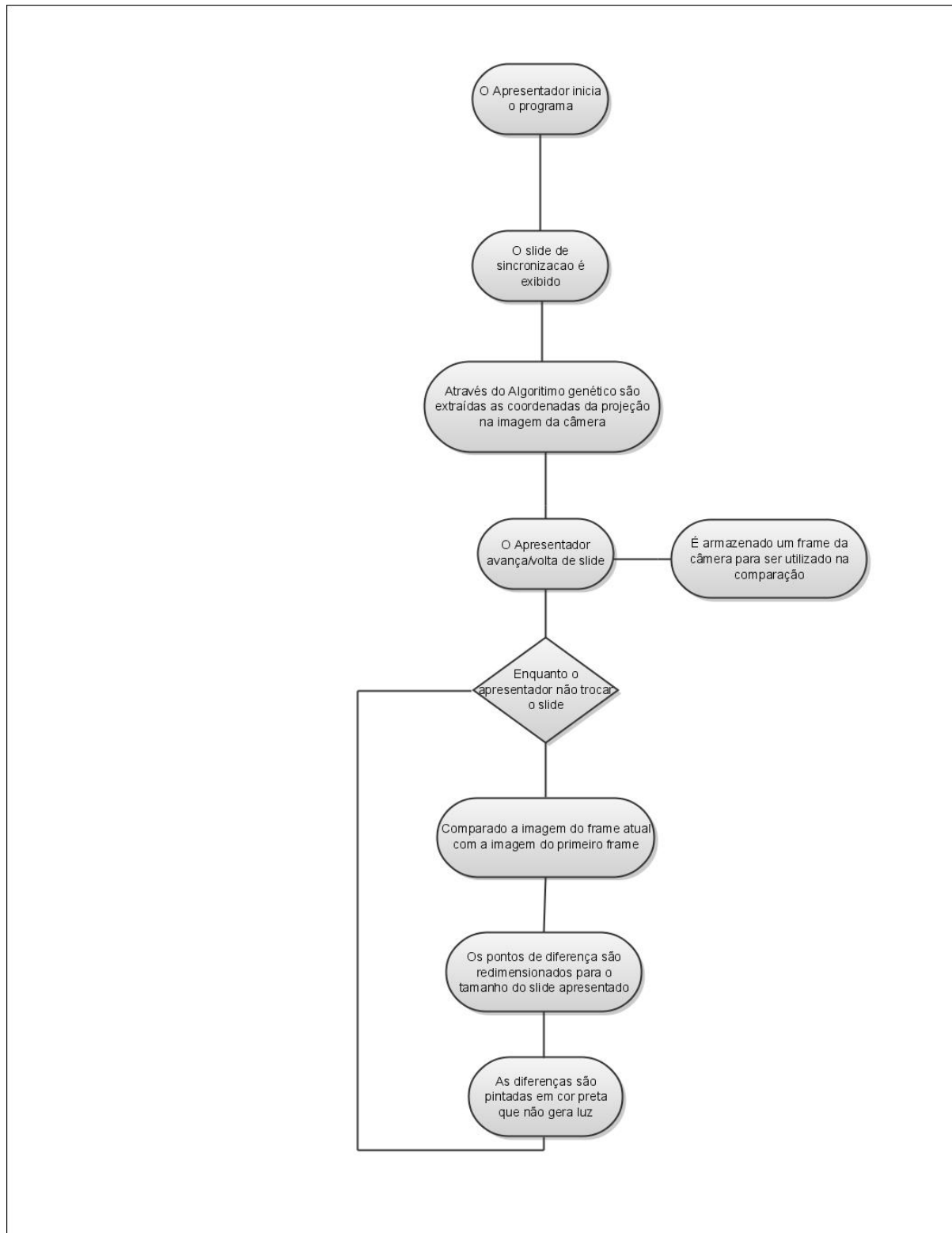


Diagrama 1 – Fluxograma do funcionamento base do programa

Slide de sincronização: É um slide em branco que facilita o algoritmo genético na busca da melhor área de projeção.

Algoritmo Genético: Explicado anteriormente.

Primeiro Frame da Câmera - Imagem retirada logo após o apresentador mudar de slide. Será usado na comparação. Uma "foto" do slide na visão da webcam.

Frame Atual da Câmera: Imagem de câmera que será usada na comparação com o primeiro frame. Esta imagem pode ou não conter uma interferência do apresentador.

Redimensionamento das Diferenças: Uma proporção entre o tamanho do slide e o tamanho da imagem de projeção da webcam. Com os pontos de diferença obtidos na comparação são pintadas as diferenças, de acordo com a proporção, no slide.

6.1 ARQUITETURA DO SISTEMA

Por se tratar de um aplicativo para a apresentação de slides, nosso primeiro requisito era desenvolver uma aplicação desktop (Swing), pois mesmo com o avanço da internet, ainda nem todos os auditórios e salas de aula estão providos com acesso à grande rede mundial. Como escolhemos a linguagem Java que é uma robusta linguagem de programação, fizemos com que nosso aplicativo fosse organizado seguindo os conceitos da orientação a objetos, evitando assim a duplicação de código e ganhando mais agilidade no desenvolvimento.

Também para que pudéssemos concentrar nossos esforços na comparação da imagem e na sincronia do aplicativo (para que seu processamento fosse feito em tempo real) utilizamos diversas bibliotecas livres como o JGAP e o JarFrameGrabber.

6.2 CASOS DE USO

6.3 DIAGRAMAS DE CASO DE USO

6.3.1 UC001 – Iniciar Apresentação

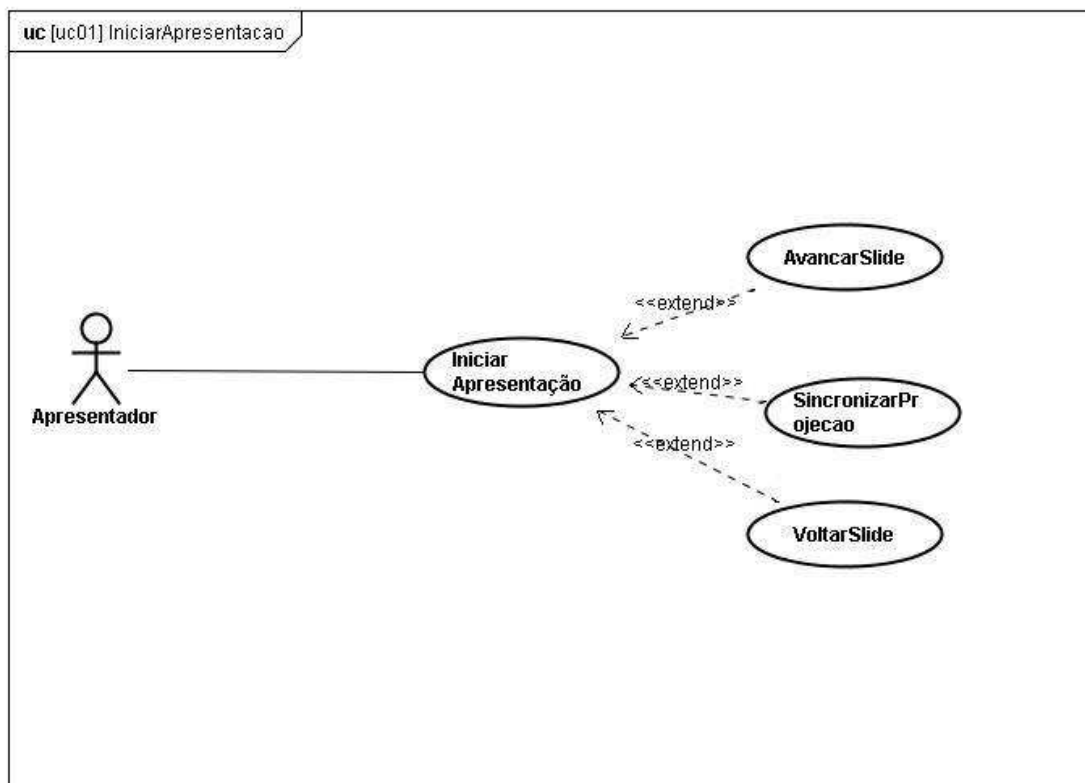


Diagrama 2 – Caso de uso – Iniciar apresentação

6.3.1.1 Descrição

Este Caso de Uso realiza a função de iniciar o aplicativo de apresentação

6.3.1.2 Pré-Condições

1. Haver instalado no computador um dispositivo de captura de vídeo

6.3.1.3 Pós-Condições

1. Sistema deve ter armazenado as coordenadas da projeção bem como as imagens que serão utilizadas como slide

6.3.1.4 Ator primário

Apresentador

6.3.1.5 Atores secundários

Sistema

6.3.1.6 Fluxo Principal de Eventos

1. O apresentador clica no botão que executa o programa.
2. O sistema exibe a tela da apresentação
3. O sistema carrega as imagens localizadas na pasta de apresentação (R2) (E1)
4. O sistema inicializa o caso de Uso [UC04] SincronizarSlide
5. O apresentador clica no botão “AvancarSlide”. (A1)(A2)
6. O sistema inicializa o caso de Uso [UC02] AvancarSlide.
6. O caso de Uso é finalizado

6.3.1.7 Fluxos Alternativos

A1. Apresentador clica no botão “Voltar Slide”

1. O sistema inicializa o caso de Uso [UC03] VoltarSlide
2. Caso de Uso é finalizado.

A2. Apresentador clica no botão “Sincronizar Slide”

1. O sistema inicializa o caso de Uso [UC04] SincronizarSlide
2. Caso de Uso é finalizado.

6.3.1.8 Fluxo de Exceção

Exceção acionada quando não houver arquivos de imagem na pasta. O programa irá alertar o usuário e finalizar a sua execução.

6.3.2 UC002 – Avançar Slide

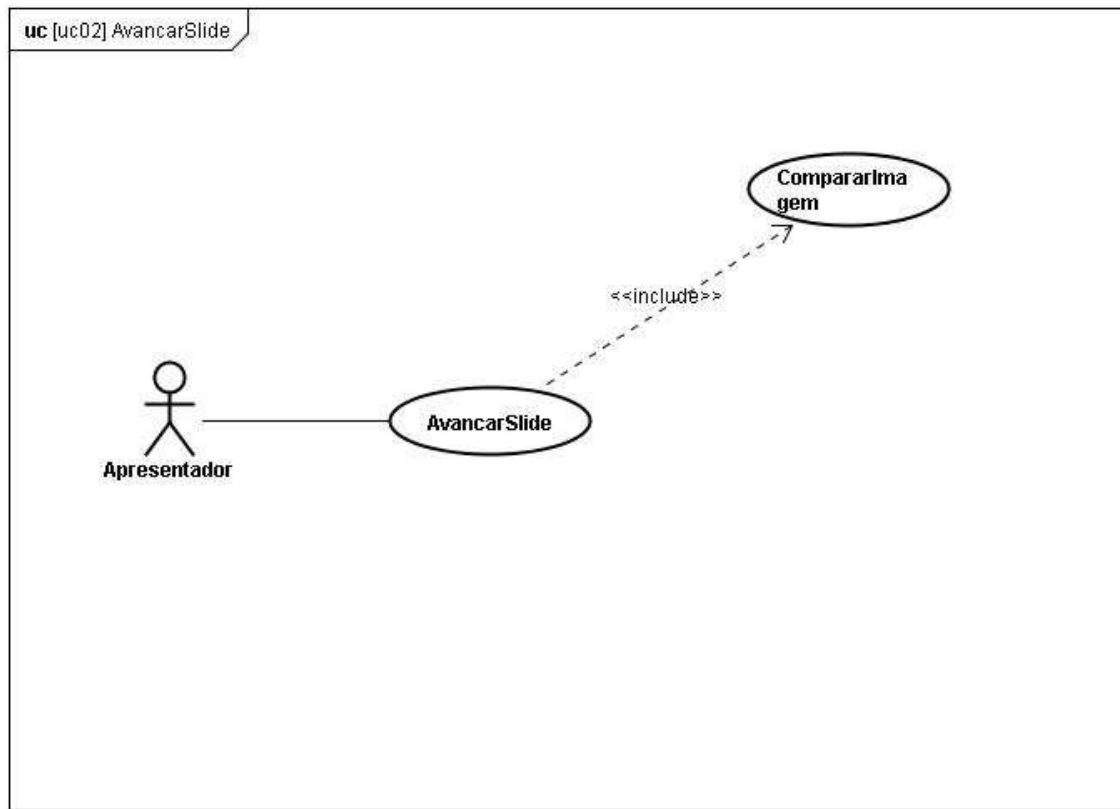


Diagrama 3 – Caso de uso – Avançar slide

6.3.2.1 Descrição

Este Caso de Uso realiza a função de exibir o próximo slide da apresentação.

6.3.2.2 Pré-Condições

1. Sistema deve ter armazenado as coordenadas da projeção e as imagens (slides)

6.3.2.3 Pós-Condições

1. Sistema deve ter trocado o slide da tela.

6.3.2.4 Ator primário

Apresentador

6.3.2.5 Atores secundários

Sistema

6.3.2.6 Fluxo Principal de eventos

1. O apresentador clica no botão “Avançar Slide”.
2. O sistema exibe o próximo slide da lista. (E1)
3. O sistema obtém e armazena a imagem do slide atual na visão da câmera (R5)
4. O sistema compara o frame do slide atual com o frame do slide armazenado.
5. O caso de Uso é finalizado

6.3.2.7 Fluxo de Exceção

Fluxo acionado quando o botão de avançar slide é pressionado no ultimo slide. O programa irá ignorar a tentativa e continuar a execução no fluxo principal

6.3.3 UC003 – Voltar Slide

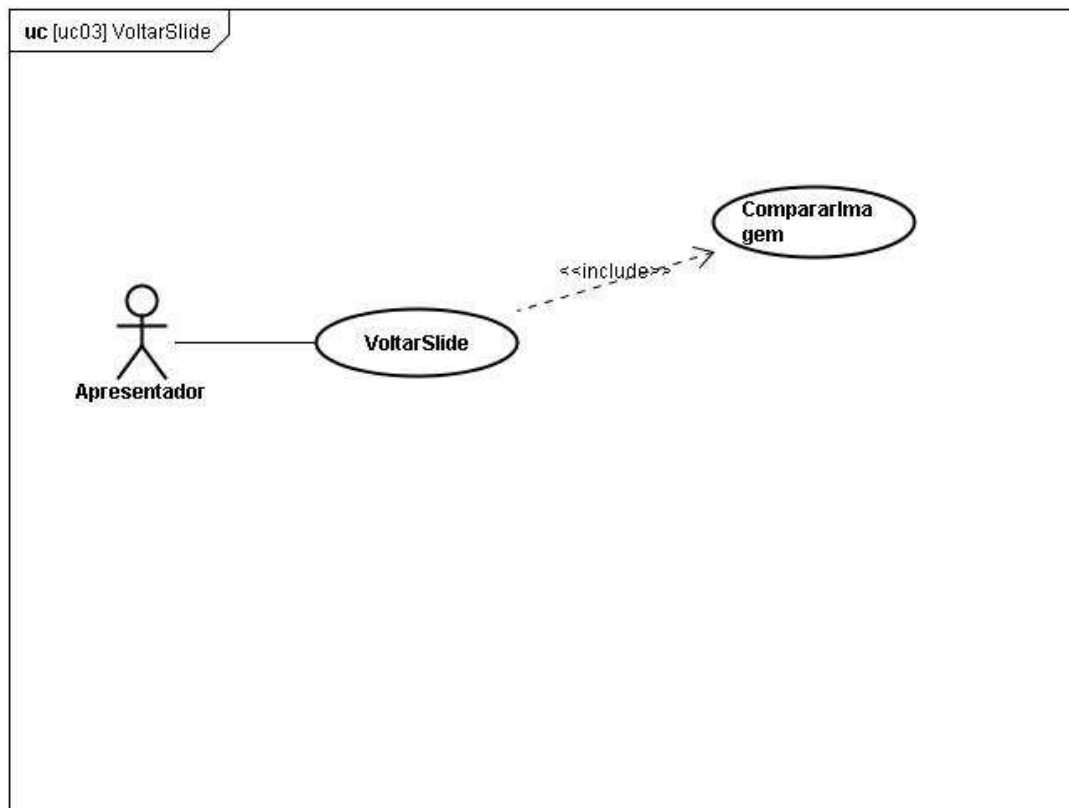


Diagrama 4 – Caso de uso – Voltar slide

6.3.3.1 Descrição

Este Caso de Uso realiza a função de exibir o slide anterior da apresentação.

6.3.3.2 Pré-Condições

1. Sistema deve ter armazenado as coordenadas da projeção e as imagens (slides)

6.3.3.3 Pós-Condições

1. Sistema deve ter trocado o slide da tela.

6.3.3.4 Ator primário

Apresentador

6.3.3.5 Atores secundários

Sistema

6.3.3.6 Fluxo Principal de eventos

1. O apresentador clica no botão “Voltar Slide”.
2. O sistema exibe o slide anterior da lista. (E1)
3. O sistema obtém e armazena a imagem do slide atual na visão da câmera (R5)
4. O sistema compara o frame do slide atual com o frame do slide armazenado.
5. O caso de Uso é finalizado

6.3.3.7 Fluxo de Exceção

Fluxo acionado quando o botão de voltar slide é pressionado no primeiro slide. O programa irá ignorar a tentativa e continuar a execução no fluxo principal

6.3.4 UC004 – Sincronizar Projeção

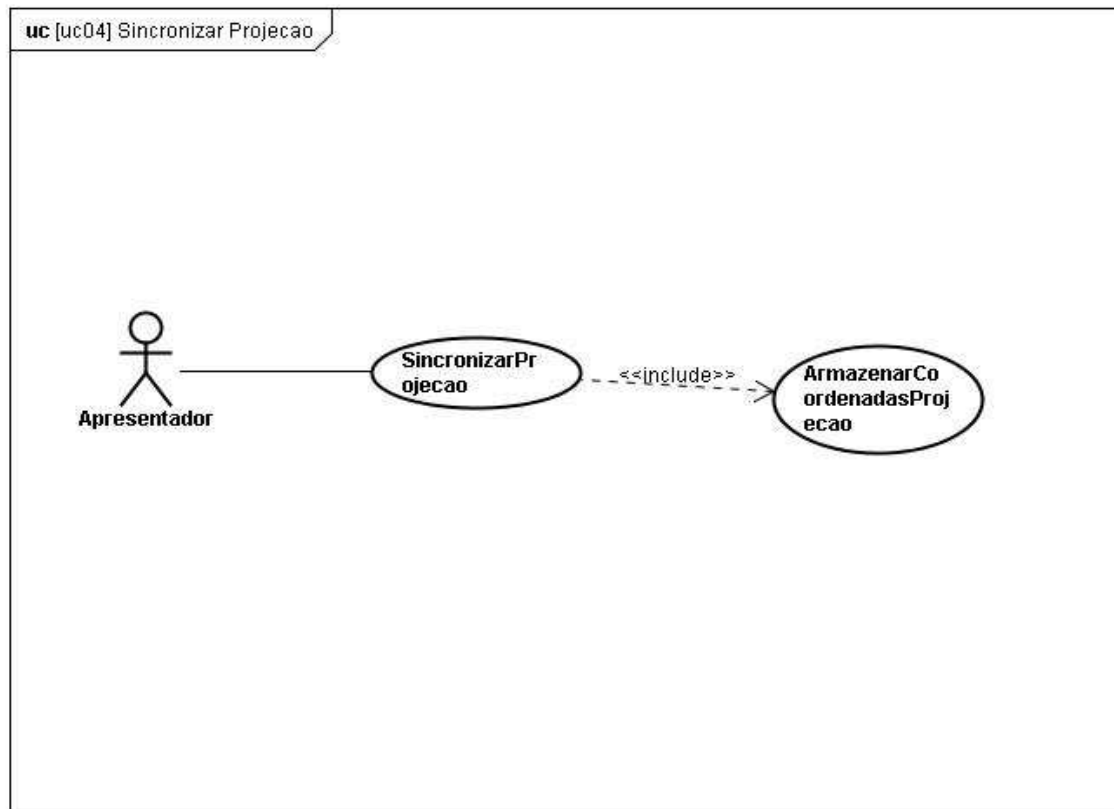


Diagrama 5 – Caso de uso – Sincronizar projeção

6.3.4.1 Descrição

Este Caso de Uso realiza a função de armazenar as coordenadas da projeção.

6.3.4.2 Pós-Condições

1. Sistema deve ter armazenado as coordenadas da projeção em relação ao slide.

6.3.4.3 Ator primário

Apresentador

6.3.4.4 Atores secundário

Sistema

6.3.4.5 Fluxo Principal de eventos

1. O sistema exibe uma projeção em branco
2. O sistema obtém um frame da câmera (R3) (R4)
3. O sistema obtém e armazena as coordenadas da projeção em relação ao slide (R1)
4. O Caso de Uso é finalizado

6.4 REGRAS DE NEGÓCIO

R1. Obtenção das coordenadas da projeção

A obtenção das coordenadas ocorre em um algoritmo genético que avalia qual a região da imagem extraída onde existe a maior concentração de pixels claros e retorna suas coordenadas.

R2. Carregar as imagens da pasta de apresentação

O sistema atribui por padrão a pasta "C:\\Apresentacao". Ele irá carregar todos os arquivos de imagem no diretório e armazenar.

R3. Câmera

Para a captura de frames deve ser utilizada uma webcam de 1.3 megapixels, ou superior.

R4. Obtenção de frames

Para a obtenção dos frames há diversas API's em Java que controlam a webcam, provendo métodos necessários para a manipulação.

R5. Obtenção do slide atual na visão da câmera

Esta imagem consiste em um frame da câmera recortado nas coordenadas da projeção, localizadas no início da execução do programa.

R6. Comparação de imagens

O sistema parte da premissa que a diferença entre o primeiro frame de um slide e os próximos será o apresentador. Então a técnica mais adequada para comparação é a subtração de imagens.

6.5 DIAGRAMAS DE SEQUÊNCIA

6.5.1 DS001 – Iniciar Apresentação

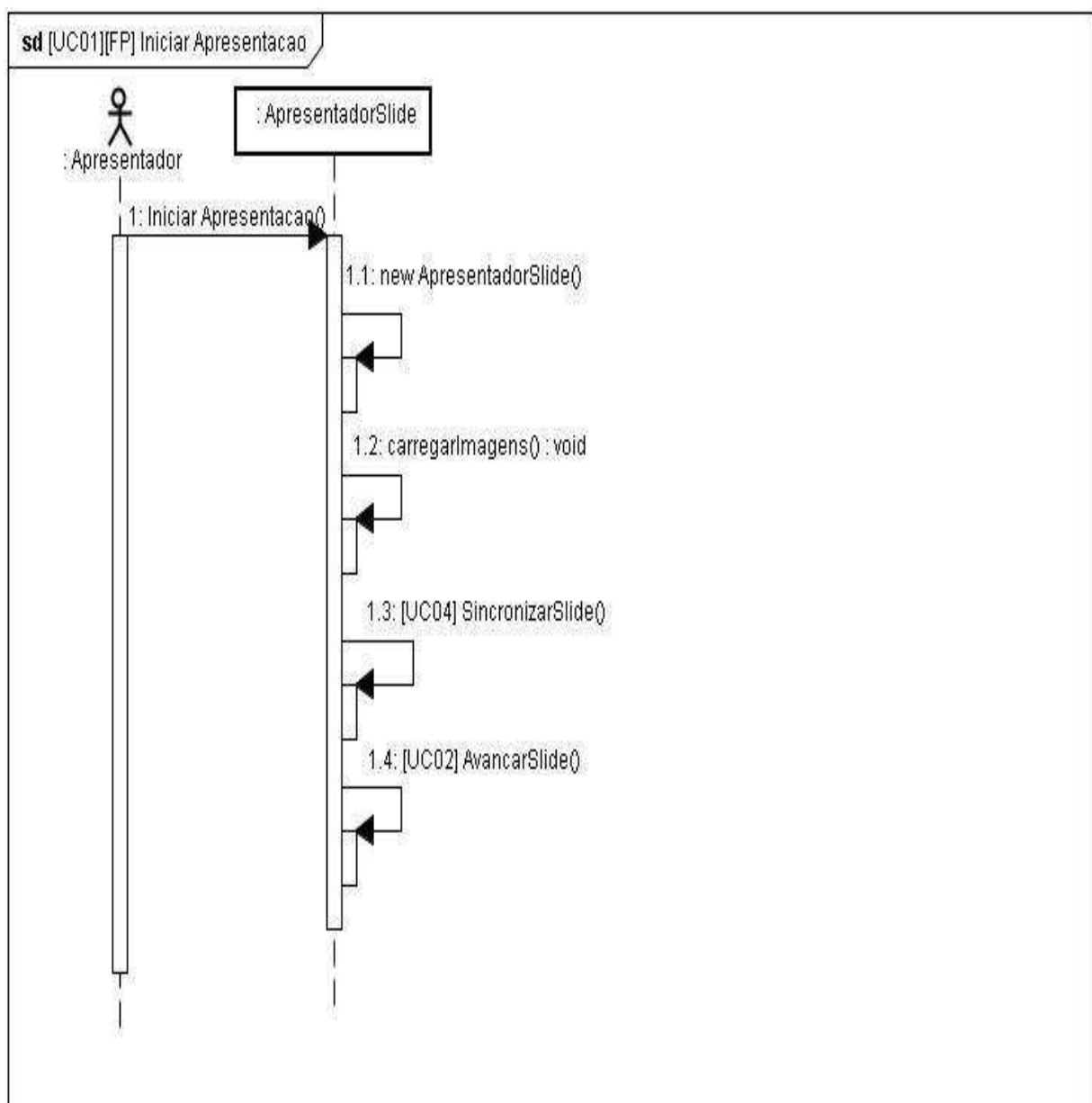


Diagrama 6 – Diagrama de sequência – Iniciar apresentação

6.5.2 DS002 – Avançar Slide

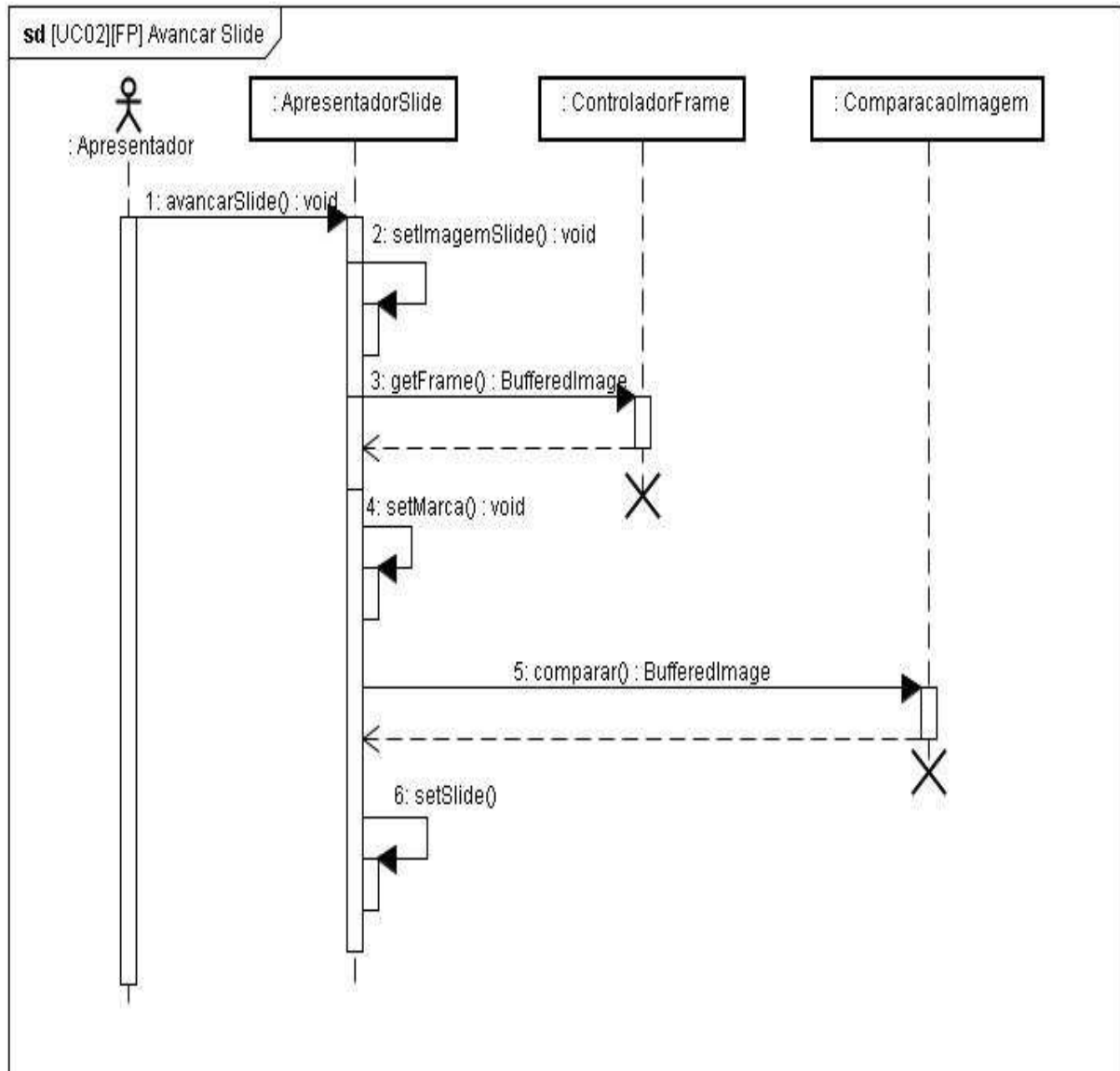


Diagrama 7 – Diagrama de sequência – Avançar slide

6.5.3 DS003 Voltar Slide

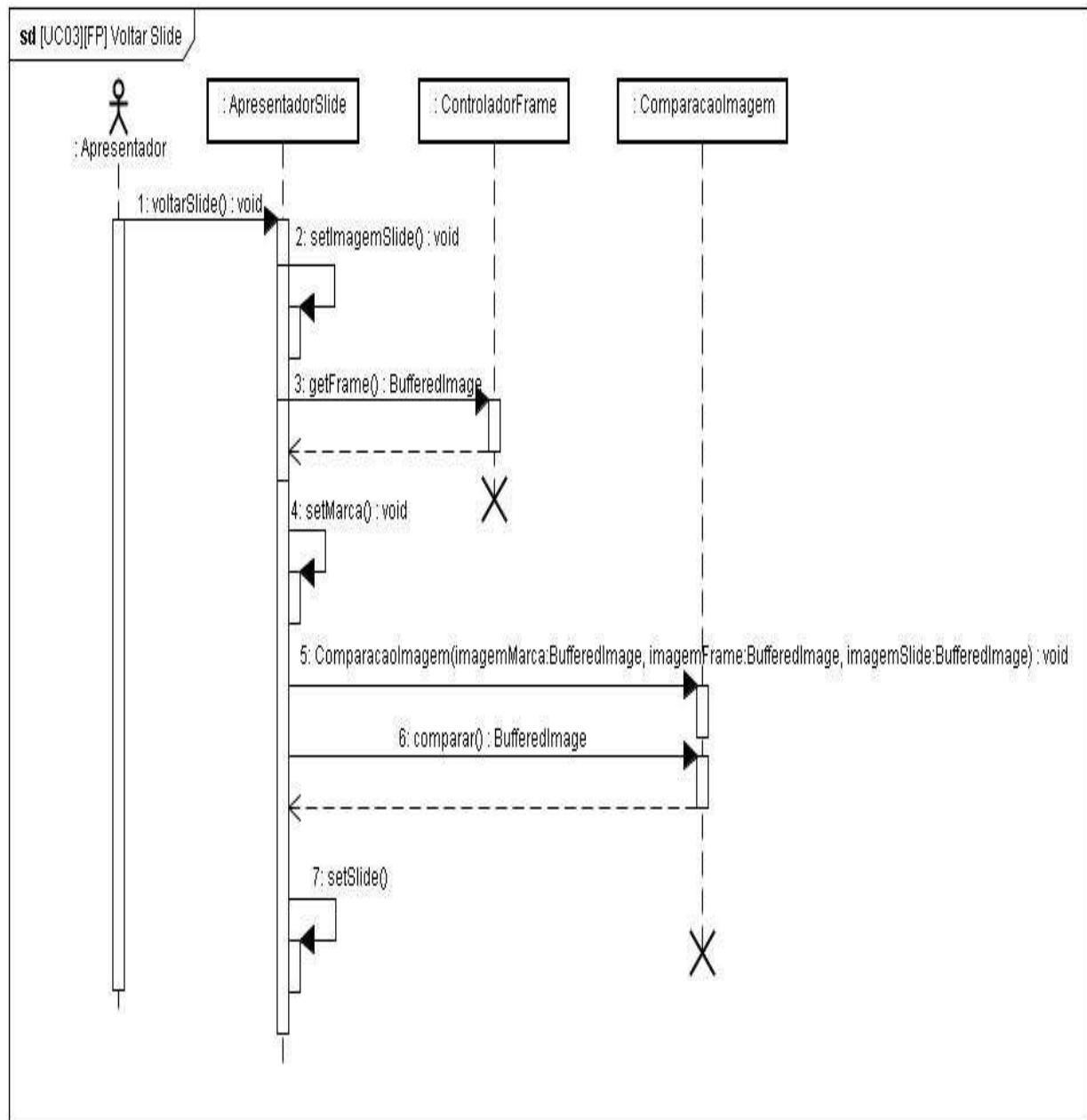


Diagrama 8 – Diagrama de seqüência – Voltar slide

6.5.4 DS004 Sincronizar Slide

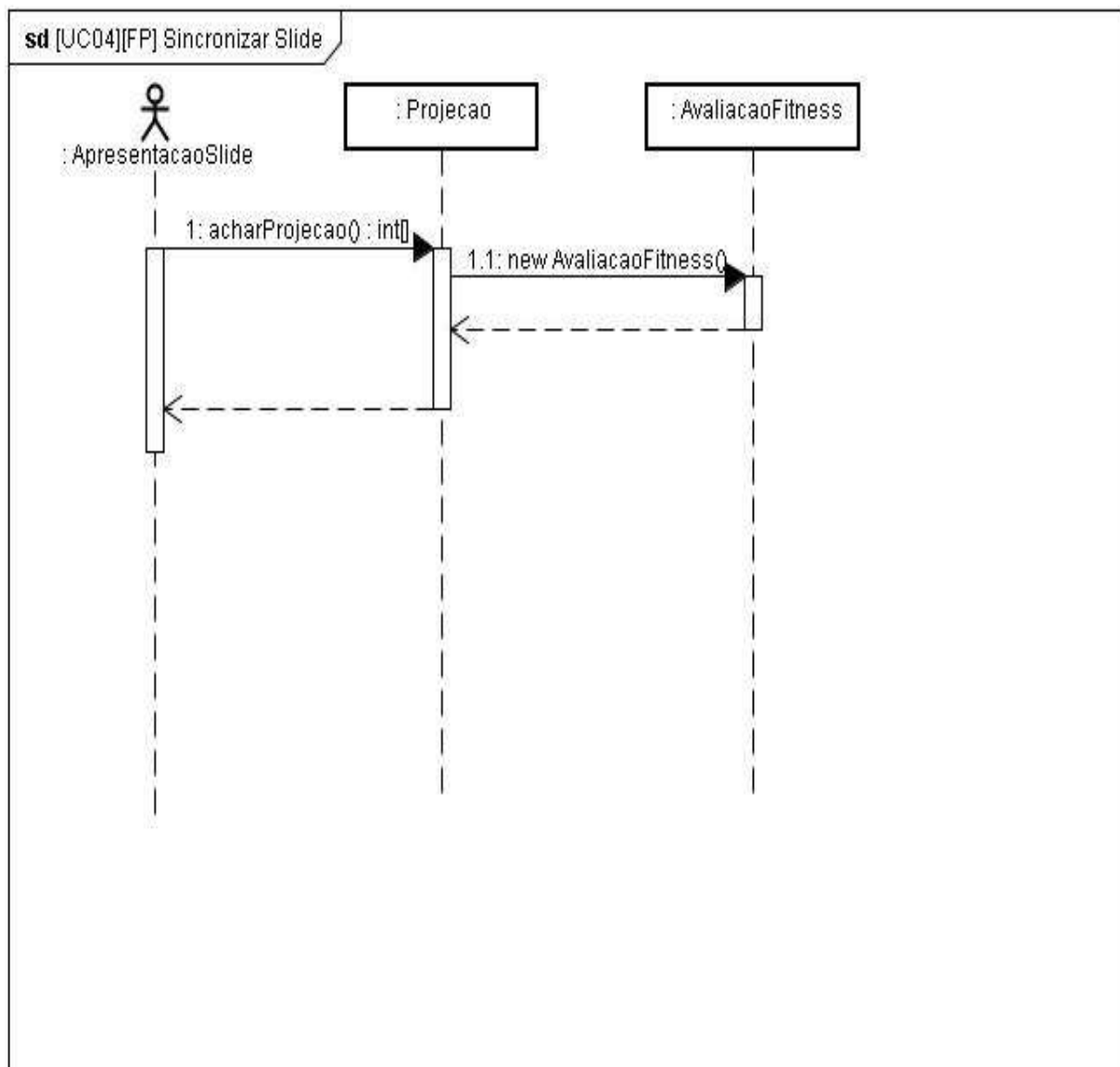


Diagrama 9 – Diagrama de sequência – Sincronizar slide

6.6 DIAGRAMA DE CLASSES

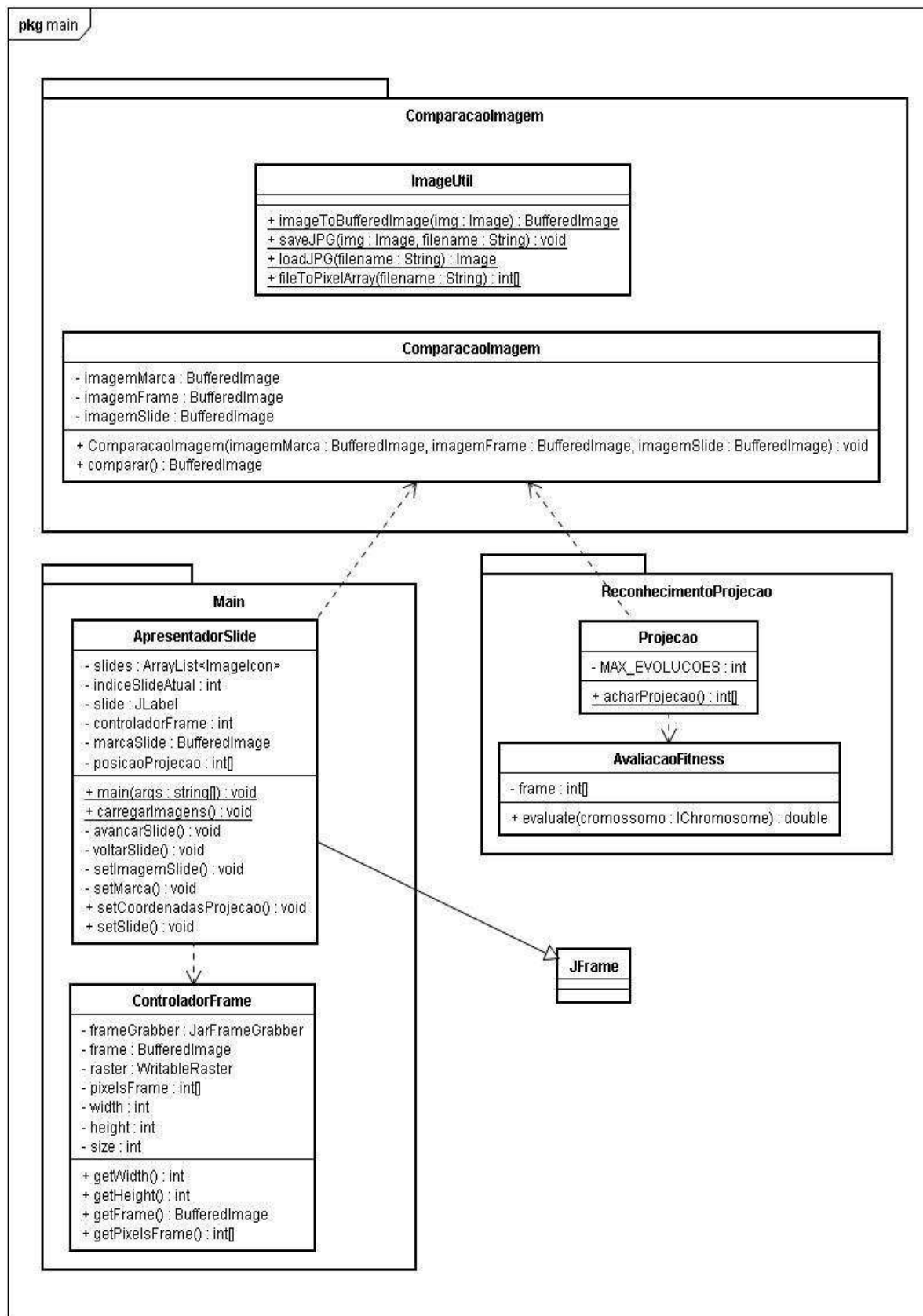


Diagrama 10 – Diagrama de Classes

7 CONCLUSÃO E TRABALHOS FUTUROS

O desenvolvimento do Aplicativo para Ergonomia de Apresentações trouxe grande aprendizado para os membros da equipe, tanto didático quanto aumento da capacidade de se trabalhar em grupos. A princípio, a impressão que tivemos foi de que não iríamos conseguir atingir o objetivo, que era emitir luz de cor preta no corpo e rosto do apresentador que interferisse no feixe de apresentação do datashow, assim simulando ausência de luz no mesmo. Porém, o objetivo foi alcançado, o aplicativo foi implementado e atingiu suas metas, e o árduo trabalho tornou-se motivo de grande satisfação.

Com este trabalho de conclusão de curso, colocamos em prática o que aprendemos durante estes 3 anos que estivemos na faculdade e ainda obtivemos novos conhecimentos através de pesquisas e reuniões com pessoas especialistas no assunto. A junção dos conhecimentos de todos os membros, aliados aos das fontes pesquisadas tornou o projeto possível e real.

No futuro, nosso aplicativo poderá contar com várias outras funções que ainda não foram implementadas. Um exemplo de função a ser desenvolvida pode ser a integração com aplicativos para criação e exibição de apresentações, como o Power Point da Microsoft e o Impress do pacote BrOffice. Outra função que poderia ser implementada é a de integração com placa de vídeo.

8 ANEXOS

8.5 GUIA DE INSTALAÇÃO

8.5.2 Requisitos Mínimos

Para o pleno funcionamento do sistema, é requerido um microcomputador com as seguintes configurações mínimas:

- Processador com velocidade superior a 1.0 GHz;
- Memória de 1 gigabyte;
- Espaço livre em disco de 50 megabytes;
- Datashow com resolução mínima de 800x600 e 2000 lumens;
- Webcam com resolução de 1.3 megapixels;

8.5.3 Passos para Instalação

Para que este aplicativo possa ser compilado é necessário seguir os seguintes passos:

- Obtenha a versão mais atual da Java Virtual Machine(JVM) em http://www.java.com/pt_BR/download/index.jsp
- Obtenha a versão mais atual do Java Developer Toolkit(JDK) em <http://java.sun.com/javase/downloads/widget/jdk6.jsp>
- Obtenha a versão Galileo do Eclipse em <http://www.eclipse.org/downloads/>
- Obtenha os arquivos fonte do JarToolkit em http://sourceforge.net/projects/jartoolkit/files/jartoolkit/jARToolKit%202.0/jARToolKit_v2.0.zip/download
- Obtenha os arquivos do JGAP em <http://sourceforge.net/projects/jgap/files/>
- Vá até o diretório onde a JVM foi instalada(C:\Arquivos de programas\Java\jre6\) e adicione na pasta lib\ext\ o arquivo 'JarToolkit.jar' e o arquivo 'Jgap.jar'
- Vá até o diretório onde a JVM foi instalada (C:\Arquivos de programas\Java\jre6\), acesse a pasta bin e adicione os arquivos “.dll”.
- Vá até o diretório do Windows e adicione os arquivos “.dll” no system32.
- Execute o aplicativo Aplicativo.jar.

9 REFERÊNCIAS

- [1] RUSS JOHN C. **The Image Processing Handbook**. Second Edition: CRC Press, Boca Raton, FL, USA, 1995.
- [2] GONZALEZ, Rafael C.; Woods, Richard E.. **Processamento de Imagens Digitais**. São Paulo, SP: Editora Edgard Blücher Ltda, 2000.
- [3] NEWMAN, Alexander. **Usando Java**. Rio de Janeiro: Campus, 1997.
- [4] Info Ciência, acessado em 21/10/2009
<http://www.infociencia.info/artigos/ARTc225.pdf>
- [5] Object Management Group, Acessado em 22/10/2009
<http://www.omg.org>
- [6] BEZERRA, Eduardo. **Princípios de Análise e Projeto de Sistemas com UML**. Ed. Campus. Rio de Janeiro, 2002.
- [7] BOOCH, Grady; RUMBAUCH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Ed. Campus, Rio de Janeiro. 2000.
- [8] Centro de Informática da UFPE, Acessado em 21/10/2009
http://www.cin.ufpe.br/~fmcf2/mestrado/ES/TOOLDAY_ARC_20060627.doc
- [9] Algoritmos Genéticos: Fundamentos e Aplicações, Marcio Nunes de Miranda, Acessado em: 22/10/2009
<http://www.gta.ufrj.br/~marcio/genetic.html>

10 GLOSSÁRIO

Bytecode - é um estágio intermédio entre o código-fonte (escrito numa linguagem de programação específica) e a aplicação final, sendo a sua vantagem principal a dualidade entre a portabilidade — o *bytecode* irá produzir o mesmo resultado em qualquer arquitetura — e a ausência da necessidade do pré-processamento típico dos compiladores

Compiladas - Linguagem compilada é uma linguagem de programação, onde o código fonte nessa linguagem é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido, através de um processo chamado compilação, usando um programa de computador chamado compilador, para uma linguagem de baixo nível, como linguagem de montagem ou código de máquina.

Ergonomia - A ergonomia, *human factors* (fatores humanos) ou *human factors & ergonomics* (fatores humanos e ergonomia), expressões pelas quais é conhecida nos Estados Unidos da América, é a disciplina científica relacionada ao entendimento das interações entre seres humanos e outros elementos de um sistema, e também é a profissão que aplica teoria, princípios, dados e métodos para projetar a fim de otimizar o bem-estar humano e o desempenho geral de um sistema. Esta é a definição adotada pela Associação Internacional de Ergonomia (*International Ergonomics Association* - IEA) em 2000.

Fluxograma - representa uma seqüência de trabalho qualquer, de forma detalhada (pode ser também sintética), onde as operações ou os responsáveis e os departamentos envolvidos são visualizados nos processo.

Frame - Frame é um conjunto de quadros (imagens) ou pacotes que formam um filme por inteiro

Freeware - Programa de computador cuja utilização não implica no pagamento de licenças de uso ou *royalties*.

Full screen – Tela inteira.

Linux - É ao mesmo tempo um kernel (ou núcleo) e o sistema operacional que roda sobre ele. O kernel Linux foi criado em 1991 por Linus Torvalds, então um estudante finlandês, e hoje é mantido por uma comunidade mundial de desenvolvedores (que inclui programadores individuais e empresas como a IBM, a HP e a Hitachi), coordenada pelo mesmo Linus, agora um desenvolvedor reconhecido mundialmente.

Swing - Swing é uma API Java para interfaces gráficas. Ela é compatível com a API AWT, mas trabalha de uma maneira totalmente diferente. A API Swing procura renderizar\desenhar por conta própria todos os componentes, ao invés de delegar essa tarefa ao sistema operacional, como a maioria das outras APIs de interface gráfica trabalham.

Threshold - Limite, limiar, em nosso aplicativo: intervalo que caracteriza um pixel como uma diferença.