

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**SISTEMA DE CAPTURA E ORGANIZAÇÃO DE
CONTEÚDOS**

Nuno Manuel Pires Prelhaz

PROJECTO

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Engenharia de Software

2012

UNIVERSIDADE DE LISBOA

Faculdade de Ciências

Departamento de Informática



**SISTEMA DE CAPTURA E ORGANIZAÇÃO
DE CONTEÚDOS**

Nuno Manuel Pires Prelhaz

PROJECTO

Trabalho orientado pelo Prof. Doutor Luís Miguel Parreira e Correia

e co-orientado por Fernando Almeida

MESTRADO EM ENGENHARIA INFORMÁTICA

Especialização em Engenharia de Software

2012

Agradecimentos

Quero expressar os meus mais sinceros agradecimentos a todas as pessoas que me apoiaram, acompanharam, inspiraram e ajudaram durante todo o meu percurso académico, especialmente nesta recta final, que corresponde à elaboração da minha tese de Mestrado de Engenharia Informática, a todos eles, um muito obrigado.

Em primeiro lugar, gostaria de agradecer ao meu orientador e ao meu co-orientador, respectivamente ao Professor Luís Correia e ao Consultor Fernando Almeida, pelo acompanhamento e suporte que deram durante o desenvolvimento deste trabalho.

Gostaria de agradecer à minha família, especialmente aos meus pais e irmão pela paciência e apoio prestado, e também a todos os meus colegas e amigos com quem tive o prazer de partilhar ideias, trabalhar e divertir-me.

Por fim, o meu último e grande obrigado destina-se à Lusodata, por me ter acolhido durante o processo de estágio/projecto.

Dedicado à minha família e amigos

Resumo

Esta tese retrata a minha experiência pessoal e profissional durante o meu trabalho que efectuei na Lusodata, como projecto de engenharia informática do mestrado em Engenharia informática.

Actualmente, são poucas as ferramentas que suportam captura de imagem, via Scanner, na ferramenta em que estão a utilizar para desenvolver, Eclipse IDE , que é uma ferramenta bem conhecida actualmente e é utilizada como base de diversas aplicações conhecidas mundialmente, tais como Adobe Flash Builder, IBM Rational Application Developer, Talend Open Studio, etc. Estas aplicações têm também como base a linguagem de programação Java, pois é nesta linguagem que o Eclipse é desenvolvido.

O projecto em que estive envolvido consistiu no desenvolvimento de uma aplicação que facilita a interacção entre o utilizador e o sistema de captura de imagem. Esta aplicação está contida num plug-in para Eclipse, escrito em Java, e utiliza como ponte entre os dispositivos e a aplicação, os drivers TWAIN (Windows) e SANE (Linux). Para facilitar a comunicação com estes drivers em Java, foi utilizado a biblioteca Gnome Morena SDK. Com esta biblioteca é possível adquirir imagens de scanners, nativamente para Java, e depois serem tratadas aqui. No desenvolvimento do plug-in foi utilizado o Eclipse PDE (Plug-in Development Environment).

O projecto atual apresenta-se com as principais funções já desenvolvidas, tais como a captura de imagens, tanto de TWAIN como de SANE, tratamento de cores, salvar imagens em ficheiros, salvar imagens em bases de dados (MySQL, SQL Server), assim como a listagem de várias imagens capturadas.

Palavras-chave: Java, TWAIN, SANE, Eclipse PDE, Scanner

Abstract

This thesis portrays my personal and professional experience during my work in Lusodata, as a project for master's degree in computer engineering.

Currently, there are few tools that support capturing image, on a Scanner, within the application people use for developing or using, such as the Eclipse IDE, which is a well-known tool and is currently used as a basis-core for various applications known worldwide, such as Adobe Flash Builder, IBM Rational Application Developer, Talend Open Studio, etc. These applications are also based on the Java programming language because this language is how Eclipse is developed.

The project I was involved in was the development of an application that facilitates interaction between the user and the image capture systems. This application is contained in a plug-in for Eclipse, written in Java, and uses as a bridge between the devices and the application, the TWAIN (Windows) and SANE (Linux). To facilitate communication with these drivers in Java, the library used was Gnome Morena SDK. With this library, you can acquire images from scanners, Java natively, and then be treated here. For the developing of the plug-in it was used Eclipse PDE (Plug-in Development Environment).

The current project is presented with the main functions already developed, such as capturing images, both as TWAIN or SANE, color treatment, save images to files, save images in databases (MySQL, SQL Server) as well as the listing of multiple images captured.

Keywords: Java, TWAIN, SANE, Eclipse PDE, Scanner

Glossário

- ECM (Enterprise Content Management)
- PDE (Plug-in Development Environment)
- IDE (Interface Development Environment)
- RUP (Rational Unified Process)
- TWAIN (Technology Without An Interesting Name (Não oficial), Image Scanning Driver)
- SANE (Scanner Access Now Easy)
- DLL (Dynamic Link Library)

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Contexto	1
1.1.1	Académico.....	1
1.1.2	Empresarial.....	1
1.2	A Lusodata.....	2
1.3	Integração	2
1.3.1	Integração na Empresa	2
1.3.2	Integração no Projecto.....	3
1.4	Motivação	3
1.5	Objectivos.....	3
1.6	Contribuições.....	4
1.7	Estrutura do documento.....	4
Capítulo 2	O Trabalho.....	5
2.1	Contexto Subjacente	5
2.2	Metodologia.....	6
2.3	Planeamento	7
Capítulo 3	Trabalho Realizado	9
3.1	Ferramentas Utilizadas	9
3.2	Trabalho Relacionado.....	10
3.2.1	jTwain e Morena	10
3.2.2	MMS Computing SDK	10
3.2.3	JavaJeff Twain.....	11
3.2.4	Comentário	11
3.3	Análise do Problema.....	11
3.3.1	Desafios.....	11
3.3.2	Arquitectura.....	12
3.4	Desenho da Solução	13

3.4.1	Casos de Uso	13
3.4.2	Diagramas de Actividade	14
3.4.3	Diagramas de Sequência	16
3.4.4	Modelo de Classes.....	19
3.5	Implementação.....	20
3.6	Testes	27
3.7	Implantação	27
Capítulo 4	Conclusões	28
4.1	Sumário.....	28
4.2	Comentário Crítico	28
4.3	Trabalho Futuro	29
Bibliografia	30
Apêndice	32

Lista de Figuras

Fig. 1: Logótipo Lusodata, pág. 2

Fig. 2: Gráfico da Metodologia RUP, pág. 6

Fig. 3: Planeamento, pág.8

Fig. 4: Arquitectura, pág. 12

Fig. 5: Casos de Uso, pág. 13

Fig. 6: Diagrama de Actividade de Digitalização de imagem e Gravação de imagens em base de dados, pág. 15

Fig. 7: Diagrama de Sequência de Consulta/Altera Opções de Captura e Digitalizar Imagem, pág. 17

Fig. 8: Diagrama de Sequência de Consulta de Listagem de imagens e Aplicação Zoom In/Out, pág. 18

Fig. 9: Diagrama de Sequência de Consulta de Gravação de imagens, pág. 19

Fig.10: Modelo de Classes, pág. 20

Fig.11: Exemplo plugin.xml, pág. 21

Fig.12: Exemplo build.properties, pág. 21

Fig.13: Plug-in implementado no Eclipse, pág. 22

Fig.14: Código do método captura de imagem, pág. 23

Fig.15: Exemplo de listagem múltipla de imagens, pág. 24

Fig.16: Código do método de gravação em ficheiro, pág. 25

Fig.17: Código do método de gravação em base de dados, pág. 26

Lista de Tabelas

Tabela 1: Classes Principais e respectiva descrição, pág. 19

Tabela 2: Funções principais, pág. 22

Capítulo 1

Introdução

Este capítulo descreve a empresa que me acolheu para a realização da tese de Mestrado, assim como o meu processo inicial de integração, o objectivo da realização do projecto que desenvolvi, a sua contribuição e a estrutura deste documento.

1.1 Contexto

1.1.1 Académico

Este relatório foi escrito enquanto parte integrante da disciplina de Projecto em Engenharia Informática (PEI), disciplina obrigatória do último ano do Mestrado em Engenharia Informática.

A orientação da parte académica foi realizada pelo Professor Doutor Luís Miguel Parreira e Correia, docente do Departamento de Informática da FCUL, que assumiu o papel de intermediário entre a Faculdade e a Lusodata.

1.1.2 Empresarial

O projecto de Sistema de Captura e Organização de Conteúdos, foi uma ideia do meu primeiro coordenador deste projecto, Fernando Gameiro, que surgiu em Junho de 2011 pela necessidade de um cliente (CPAS) da minha empresa, Lusodata, ter dificuldade em organizar as capturas através de diversos dispositivos, no sistema actual.

Como eu já tinha terminado o primeiro ano do Mestrado em Engenharia de Informática e faltava-me apenas a oportunidade para poder completar o segundo ano através da disciplina de Projecto de Engenharia Informática, foi-me atribuída esta proposta assim como para a Faculdade de Ciências da Universidade Lisboa, Departamento de Informática.

1.2 A Lusodata



Logótipo da Lusodata (figura 1)

Constituída em 1982 a LUSODATA iniciou a sua actividade com o desenvolvimento de software específico para a Área Seguradora. Potenciando e alargando a sua posição junto do mercado, a LUSODATA adere em 1983 ao programa de Agentes então iniciado pela IBM Portuguesa, com a qual mantém até hoje uma estreita parceria.

Face ao crescimento do mercado nacional das Tecnologias de Informação a LUSODATA estabeleceu uma parceria para distribuição dos produtos da Synon Inc. – Synon/2E e Obsydian. Actualmente, CA 2E e CA Plex, esta linha de produtos de apoio ao desenvolvimento de Software é propriedade da CA. Esta abordagem e posicionamento face ao então emergente mercado do CASE deu provas do esforço e atenção permanentes da nossa companhia na antecipação da mudança - razões pela qual somos hoje Microsoft Solution Provider e parceiros nos mais variados projectos de TI.

A LUSODATA entende a verdadeira realidade do mercado empresarial e a sua dinâmica competitiva, daí o nosso investimento permanente e esforço centrado na inovação e na transmissão de valor acrescentado. Os nossos objectivos passam pela excelência na oferta de soluções com o suporte de profissionais experientes e a satisfação plena do cliente. (Fonte: [13]).

1.3 Integração

1.3.1 Integração na Empresa

Eu entrei na empresa Lusodata em Abril de 2009, começando por um projecto de estágio inicial, depois em contrato de part-time parcial para poder frequentar as aulas de primeiro ano de Mestrado em Engenharia Informática na Faculdade de Ciências da Universidade de Lisboa. No fim de terminado o primeiro ano de mestrado, fui integrado nos quadros da empresa e tenho estado envolvido em vários projectos para clientes da Lusodata desde este percurso. Foram-me atribuídos computador e e-mail empresarial para o desempenho das minhas funções.

1.3.2 Integração no Projecto

A integração neste projecto começou por meu interesse na necessidade em acabar o mestrado (2º ano), e para poder ficar na empresa Lusodata, foi-me proposto a mim e ao FCUL/DI, um projecto de estágio para a disciplina de Projecto Engenharia Informática, de modo a que pudesse concluir o meu mestrado. De início tive algum tempo disponível de investigação mas devido a estar envolvido em outros projectos na Lusodata, não pude estar dedicado a este projecto a full-time, causando impacto nos tempos previstos de projecto.

1.4 Motivação

A principal motivação neste projecto é o facto de não existir actualmente nenhum adaptador, em linguagem Java, para o tratamento de captura de imagens, que funcione facilmente em qualquer plataforma de desenvolvimento, seja ela em Alfresco, Talend, Flex ou qualquer outra plataforma desenvolvida em Eclipse. Sendo este projecto OpenSource, outra motivação vai ser a oportunidade de outras pessoas poderem contribuir para a evolução futura deste projecto.

Considerando a importância do ECM (Enterprise Content Management) nas organizações, a disponibilidade de soluções simples e intuitivas na captura e catalogação de conteúdos digitais a partir de suportes físicos, este projecto contribuirá certamente para a adopção deste paradigma na comunidade Eclipse.

A motivação mais vantajosa em relação ao projecto em si, é a vantagem e facilidade com que o programador vai ter em capturar através de qualquer scanner, seja em que ambiente for (Linux ou Windows), com as diversas opções de digitalização.

Para a Lusodata, a principal vantagem que pretende atingir com este projecto reside na utilização deste plug-in nas nossas ferramentas de trabalho, assim como nas soluções que desenvolvemos para os nossos clientes, que utilizam Eclipse como base e necessitem captura de imagem via scanner ou qualquer outro interface de captura.

1.5 Objectivos

O objectivo principal deste trabalho é facilitar aos utilizadores e programadores que utilizam qualquer aplicação em Eclipse principalmente, para rapidamente digitalizarem imagens e documentos consoante as suas necessidades e facilmente guardarem no sítio desejado, para depois poderem ser utilizadas. Outro objectivo importante reside no contributo para a comunidade tecnológica que desenvolve sobre a plataforma Eclipse. Também como objectivo importante é disponibilizar uma alternativa OpenSource às

actuais ferramentas pagas que são disponibilizadas no mercado. Esta solução actualmente implementada suporta também multiplataforma, pois corre em diversos sistemas operativos de acordo com os respectivos drivers (Windows TWAIN e Linux SANE)

1.6 Contribuições

A contribuição a atingir com a construção deste componente pode ser resumida nos seguintes pontos:

- Facilitar e simplificar o processo de interacção do utilizador com o scanner
- Permitir a visualização e controlo das imagens capturadas
- Potenciar a captura de conteúdos para alimentar as soluções ECM, em ambiente empresarial.

Outra contribuição relevante do ponto de vista técnico, resume-se na implementação destas funcionalidades na forma de Plug-in para a plataforma Eclipse, endereçando uma necessidade da comunidade e incrementando o potencial das soluções desenvolvidas sobre ou que integram com esta plataforma.

1.7 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 2 – A descrição do trabalho/projecto, objectivos, a maneira em como foi abordado, a metodologia que foi utilizada e o seu planeamento.
- Capítulo 3 – O trabalho realizado, em vários subcapítulos que descrevem todo o processo desde a análise até à implementação da aplicação.
- Capítulo 4 – Conclusões e sumarização de todo o trabalho desenvolvido, análise crítica e projecções para o futuro deste projecto.

Capítulo 2 O Trabalho

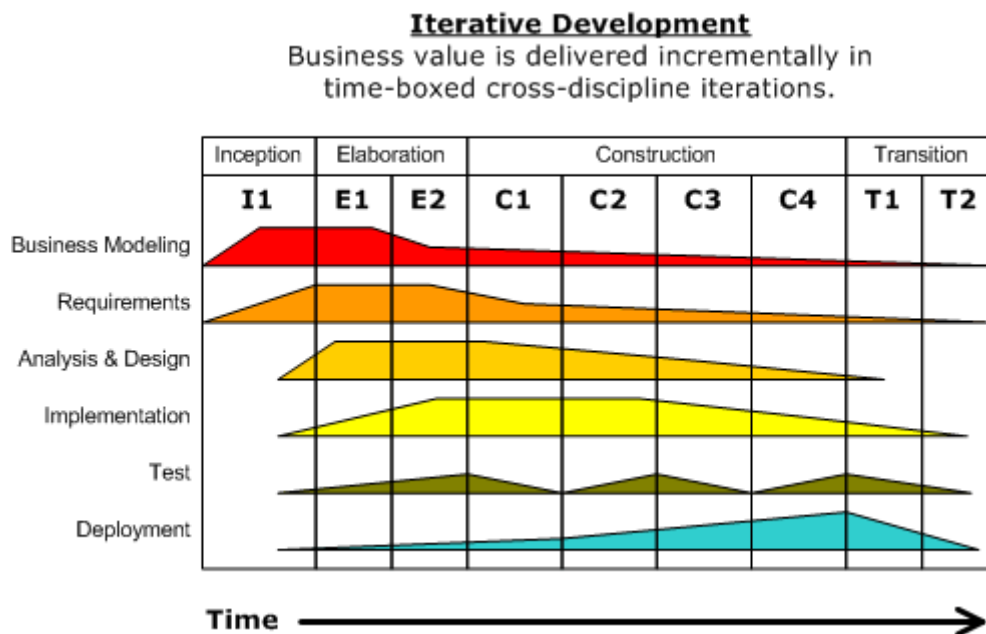
2.1 Contexto Subjacente

Este projecto foi desenvolvido para a empresa Lusodata, com o intuito de poder ser aplicado nos computadores dos nossos clientes. O contexto deste projecto consistiu, em especial, na criação de uma aplicação Java que capturasse imagens e fizesse também o tratamento destas, assim como o método de salvaguarda destas, para a empresa CPAS (Caixa de Previdência dos Advogados e Solicitadores).

2.2 Metodologia

Para o desenvolvimento deste projecto, foi utilizada a metodologia RUP [9] (Rational Unified Process), criado pela Rational Software, que consiste na abordagem de orientação a objectos na concepção e é projectado e documentado utilizando a notação

UML (*Unified Modeling Language*) para demonstrar os processos em acção. É também considerado um processo pesado, especialmente é utilizado me projectos grandes, mas pode também ser facilmente adaptado a projectos pequenos.



(Gráfico da Metodologia RUP, Fonte: [10], figura 2)

Em todas as etapas deste projecto, que são:

- Estudo no mercado
- Levantamento de requisitos
- Análise e Desenho
- Implementação
- Testes
- Implantação
- Manutenção e Suporte

São abordadas as seguintes fases: Fase de concepção, Fase de elaboração, Fase de construção, Fase de transição, durante todo o processo de desenvolvimento no projecto, descritas no Apêndice do RUP.

2.3 Planeamento

O planeamento, como está demonstrado no mapa de gantt (figura 3), é composto pelas fases clássicas de um projecto de desenvolvimento de software.

Sofreu alterações, em especial atrasos, devido ao meu estágio para este projecto não ter qualquer tipo de tempo reservado (full-time), pois estive envolvido noutros projectos para a empresa Lusodata. Estes atrasos deveram-se também com a mudança do meu antigo coordenador (Fernando Gameiro) até à proposta de um novo e actual (Fernando Almeida), dentro da empresa Lusodata. Este projecto estava previsto inicialmente ser terminado e com as funcionalidades todas implementadas, testes, implantação e suporte, em Junho (6 meses) e não era suposto ter demorado até Setembro (9 meses), mas devido a estes incidentes, pois eu também já estava inserido na empresa Lusodata e estava inserido em vários projectos, fez com que não tivesse o tempo e condições ideais para a execução deste projecto a full-time como era suposto, e portanto, foi alargado o prazo de entrega. Estive focado o máximo que pude no tempo que me foi disponível.

Foram ajustados os dias para cada fase de desenvolvimento, de forma a corresponder aos dias de entrega real.

(Figura 3 -
Planeamento)



Capítulo 3 Trabalho Realizado

3.1 Ferramentas Utilizadas

As ferramentas utilizadas foram: o Eclipse versão Helios com a versão PDE (Plug-in Development Environment), que é dedicada ao desenvolvimento de aplicações J2EE e Plug-ins para o eclipse; uma simples base de dados MySQL para testar salvar as imagens capturadas.

As tecnologias utilizadas neste projecto foram:

- Java
- Twain
- Sane
- SQL (na opção de salvar em base de dados).

As maquetes e diagramas foram desenhados no Microsoft Visio 2010. O planeamento do projecto foi desenvolvido no Microsoft Project 2010.

3.2 Trabalho Relacionado

Após a análise de trabalhos relacionados com este tema, foram encontrados alguns trabalhos semelhantes (na sua maioria comerciais) sobre a interação de aplicações com os scanners através do TWAIN [1] do Windows assim como o SANE [2] do Linux. Porém a maior parte deles são desenvolvidos em C++, visto que o TWAIN assim como SANE, são desenvolvidos em C++ também, dificultando assim, exemplos de ligação, para aplicações em Java.

Porém foram encontrados alguns trabalhos semelhantes, que fazem captura de imagem assim como algumas funcionalidades do TWAIN:

- jTWAIN / jSANE [3][4]
- Morena [5]
- MMS Computing SDK [6]
- JavaJeff Tutorial [7]

3.2.1 jTwain e Morena

Ambas estas ferramentas/bibliotecas são poderosas na captura de imagem, seja TWAIN ou SANE, trazem já uma framework desenvolvida pela Asprise e Gnome Software (Morena 6) respectivamente, com um conjunto de funcionalidades para ser utilizadas em aplicações Java, tais como tratamento de OCR, tratamento da Imagem para captura, contraste, luminosidade, etc. No entanto estas API são comerciais, apenas a API da Morena é possível adquirir uma licença de estudo que dura 5 meses.

3.2.2 MMS Computing SDK

A MMS Computing desenvolveu um SDK para Java, disponibilizado em OpenSource, em que faz também a ponte entre o Java e o Twain/Sane. No entanto esta framework desenvolvida pela MMSC, está desatualizada assim como o seu respectivo website.

Têm disponível como exemplo apenas, uma applet de teste desta framework, em que apenas apresenta a opção de seleccionar o sistema de captura de imagem a utilizar e a opção de aquisição da imagem.

3.2.3 JavaJeff Twain

Um programador conhecido na comunidade Java (java.net entre outros), chamado Jeff Friesen, publicou também em 2004 um exemplo de uma simples aquisição de imagem do driver do Twain, através do Java, fazendo também a ponte em C++ e utilizando JNI para comunicação de ambos.

3.2.4 Comentário

Todos os trabalhos relacionados não são adequados à nossa solução porque, alguns são comerciais e saem fora do âmbito do projecto (OpenSource), e os trabalhos que são livres, fazem apenas simples aquisições de imagens dos dispositivos de captura e não executam as restantes opções incluídas neste projecto e nem são utilizadas como plug-in para o Eclipse. A solução utilizada e adequada para este projecto dos trabalhos relacionados, foi a API Gnome Morena, para a captura de imagens.

3.3 Análise do Problema

Como é descrito nas contribuições deste trabalho, este projecto consiste no melhoramento e facilidade da interacção entre utilizador e os sistemas de captura, e a integração da aplicação como plug-in para a ferramenta Eclipse.

3.3.1 Desafios

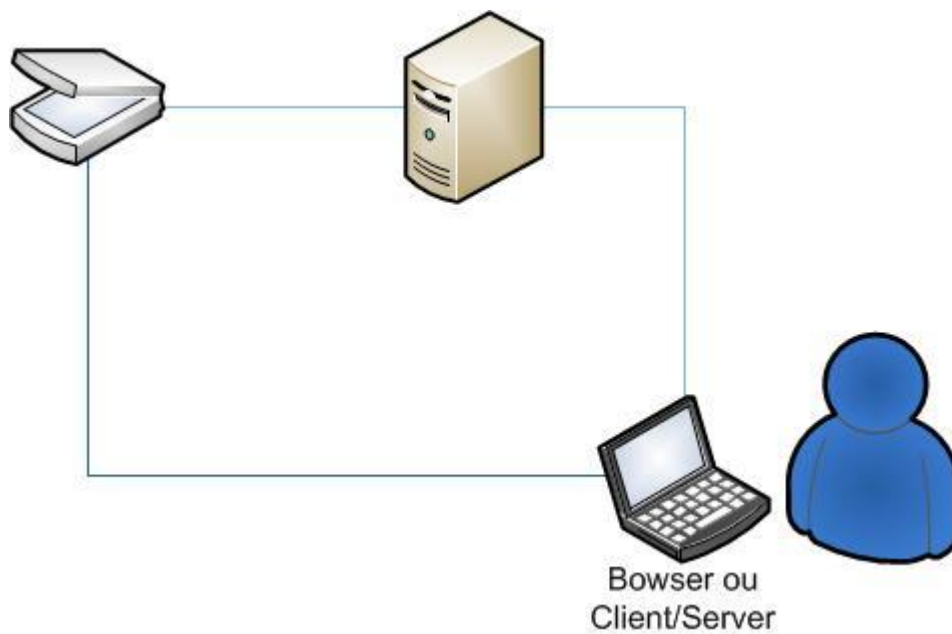
Como já foi referido na motivação, na comunidade Open Source podemos encontrar várias soluções concebidas para auxiliarem as organizações a lidar com o problema dos conteúdos. Independentemente do nível de adopção, um dos componentes mais fracos reside na usabilidade do componente de captura e visualização de imagem. Ciente destas limitações e da inexistência de Plug-ins para a plataforma Eclipse que simplifiquem e melhorem esta limitação das soluções ECM (*Enterprise Content Management*), propus-me contribuir com o meu trabalho para tentar melhorar alguns destes aspectos.

O componente que se construiu permite interagir com os scanners através de driver TWAIN II e ou SANE na captura de imagens de conteúdos, permite visualizar as

imagens obtidas, navegar num conjunto de imagens, aceitá-las e “passar” estes conteúdos à solução de ECM que for configurada.

3.3.2 Arquitectura

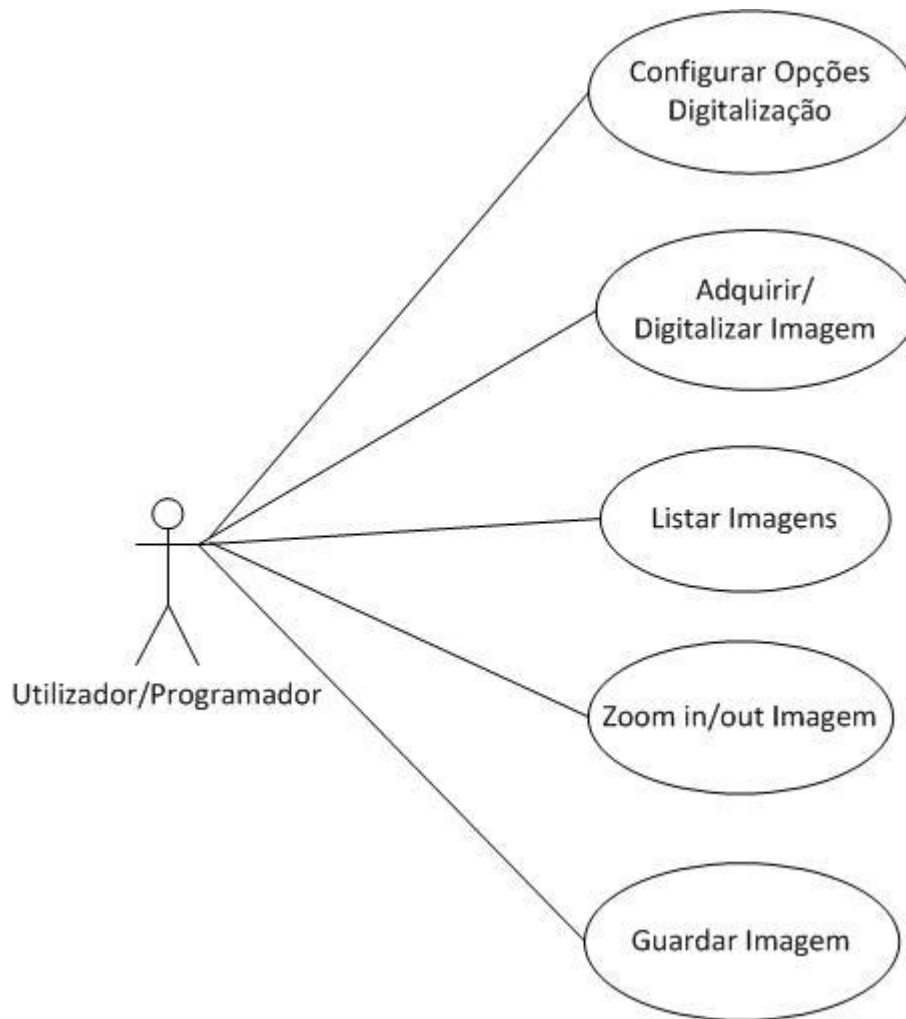
Arquitectura do projecto, composta pelo browser ou Client/Server, servidor applicacional e o dispositivo de captura de imagem (figura 4).



3.4 Desenho da Solução

3.4.1 Casos de Uso

Estas são as opções ao nível de utilizador ou programador em Eclipse, através do Plug-In. (figura 5).



3.4.2 Diagramas de Actividade

O Diagrama de actividade é um diagrama definido pela Linguagem de Modelagem Unificada (UML), e representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma actividade para outra. [11]

Foram desenhados dois diagramas de actividade (figura 5), um para o processo de digitalização de imagem e outro para gravação de imagens em base de dados.

O processo de digitalização descreve apenas que foi escolhido digitalizar a imagem.

Se o processo correu bem:

- é enviada uma mensagem de digitalização com sucesso, e volta ao inicio.

Senão:

- é enviada uma mensagem de erro e volta a tentar digitalizar.

O processo de gravação de imagens em base de dados mostra as fases desde a introdução dos parâmetros de entrada até à fase de inserção da imagem no campo da base de dados.

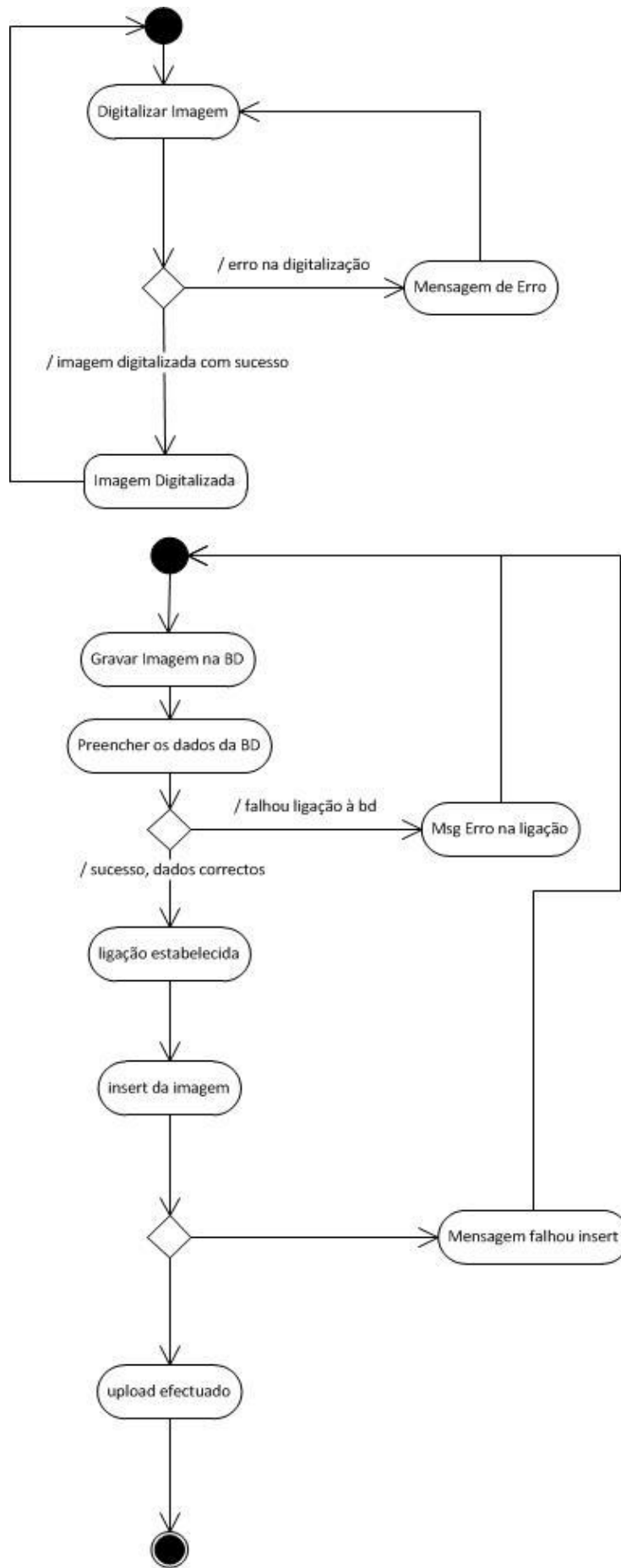


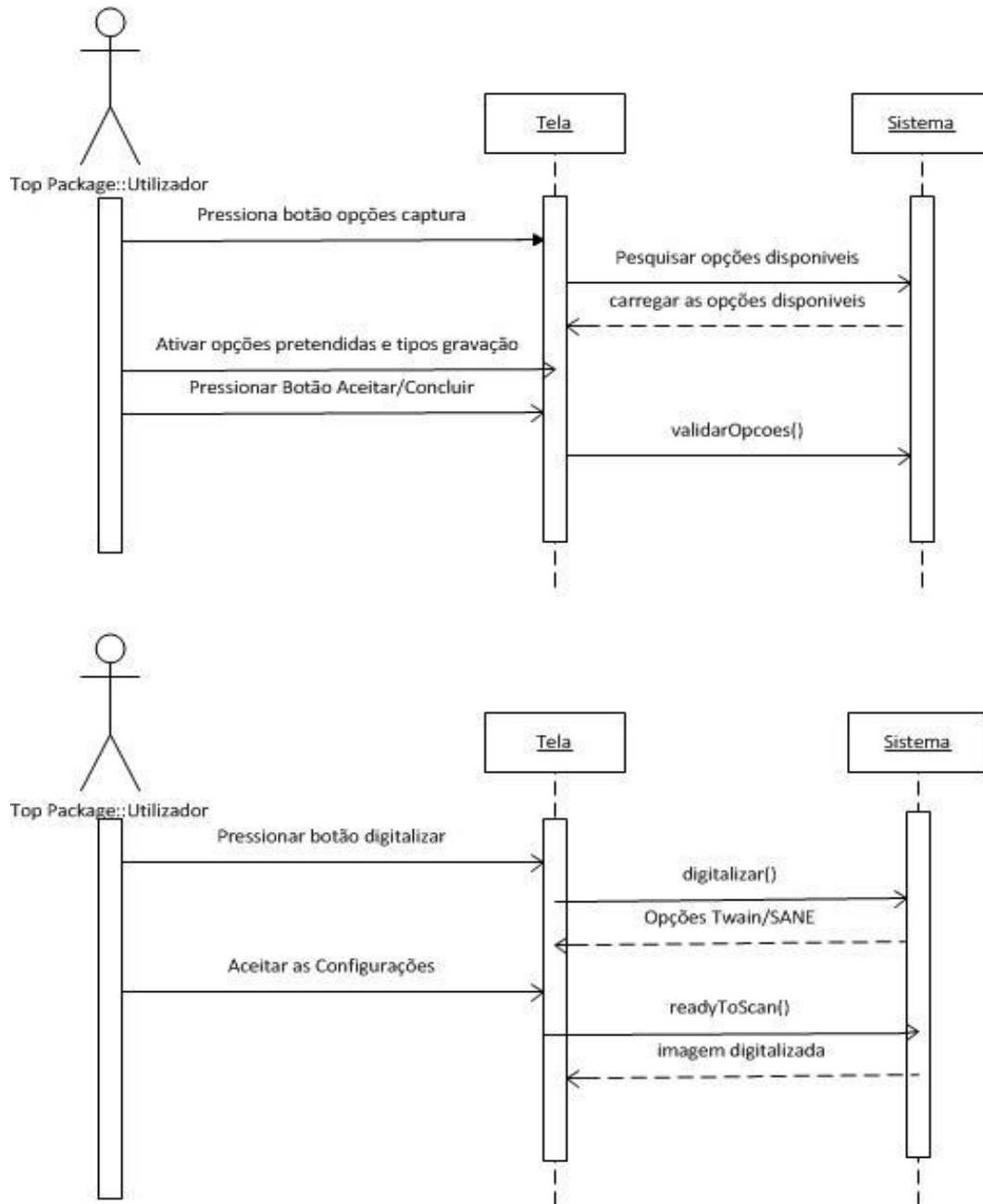
Fig.6 –
Diagrama de
Actividades

3.4.3 Diagramas de Sequência

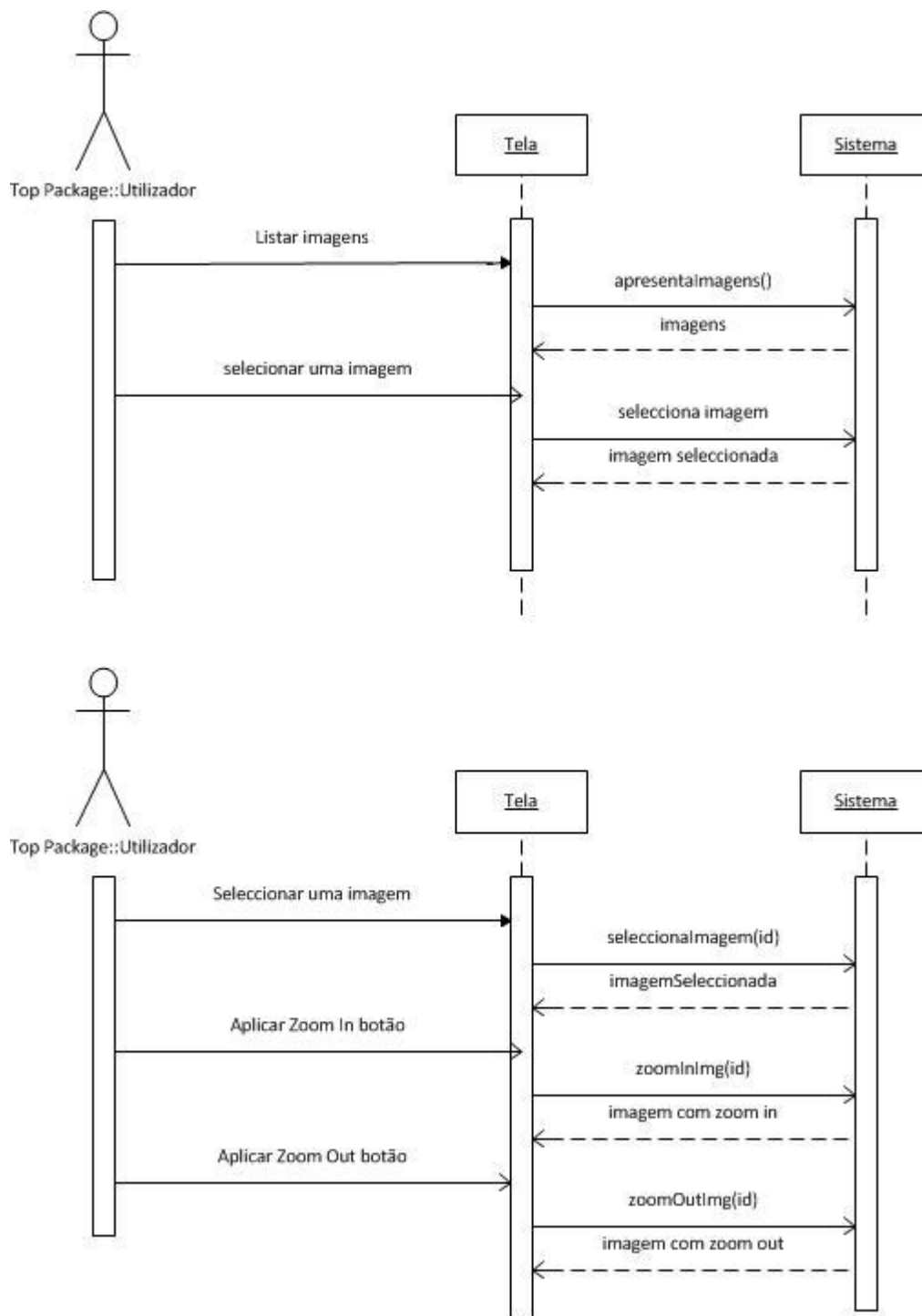
Diagrama de sequência (ou Diagrama de Sequência de Mensagens) é um diagrama usado em UML (*Unified Modeling Language*), representando a sequência de processos (mais especificamente, de mensagens passadas entre objectos) num programa de computador.[16]

Para este projecto foram desenvolvidos 5 diagramas, para os seguintes processos:

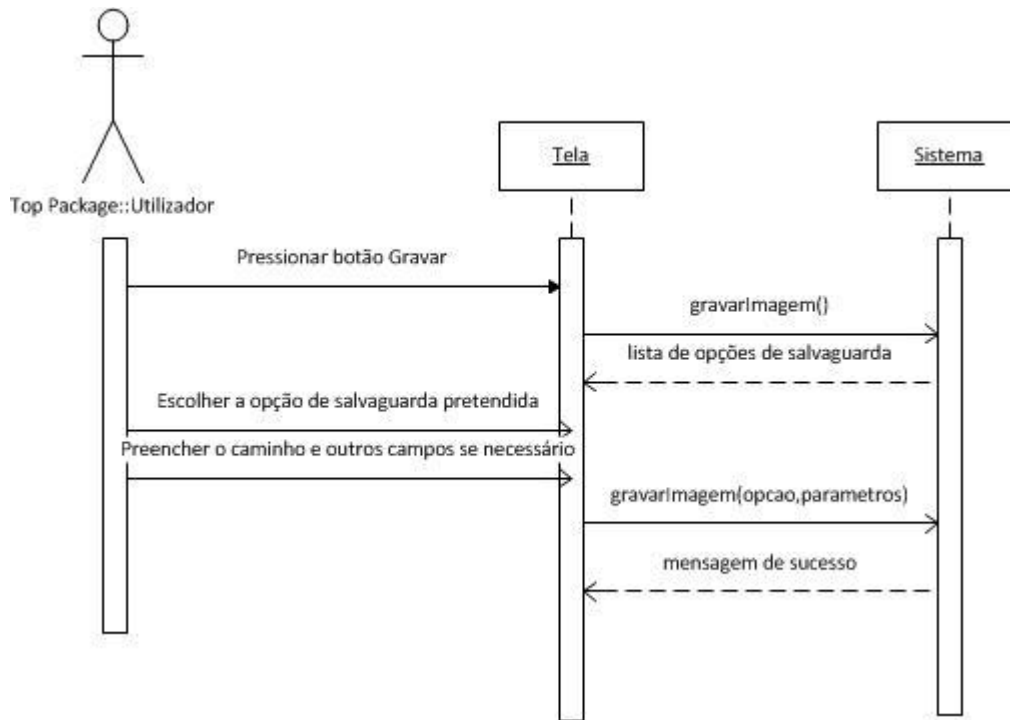
- Utilizador deseja consultar/alterar opções de captura; (fig.7)
- Utilizador deseja digitalizar uma imagem; (fig.7)
- Utilizador lista/selecciona as imagens digitalizadas; (fig.8)
- Utilizador selecciona uma imagem e aplica zoom in/out sobre ela; (fig.8)
- Utilizador guarda uma imagem em disco ou na base de dados pretendida; (fig.9)



(Fig. 7 – Diagramas de Sequência – Opções de Captura e Digitalização de Imagem)



(Fig. 8 – Diagramas de Sequência – Listagem de Imagens e Aplicação de Zoom).



(Fig.9 – Diagrama de Sequência – Gravação de imagens em Sistemas de Base de Dados/ECM)

3.4.4 Modelo de Classes

OpenLusoScannerUIAction	Esta classe contém o esqueleto utilizado pelo plug-in do Eclipse, para a opção de menu, de execução da aplicação.
LusoScannerUI	Classe que contém a interface visual para o utilizador
LusoScanner	Classe Principal, que contém todas as chamadas às funções principais
LusoScannerBD	Classe que trata da ligação às bases de dados disponíveis, assim como a gravação da imagem em campos BLOB (Binary Large Object).
LusoScannerECM	Classe que trata da ligação a um sistema ECM (neste caso, apenas foi testado o IBM Filenet mas não ficou pronto).

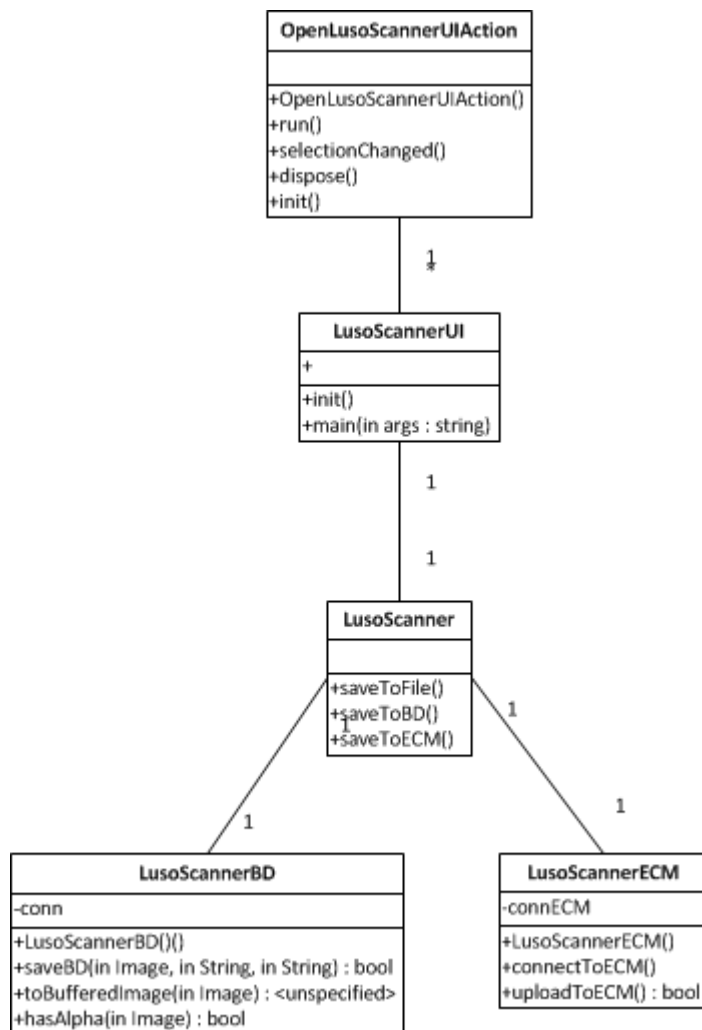


Fig.10 – Modelo de Classes

3.5 Implementação

A implementação começou, após o desenho da solução, pela criação de um projecto plug-in no Eclipse PDE, criando uma classe `OpenLusoScannerUIAction.java` dentro do package `com.example.lusodata.actions`, que contém a inicialização da classe `LusoScannerUI`, quando este menu/opção é chamada.

Na criação das opções de um plug-in para Eclipse, é necessário definir nos ficheiros gerados pelo Eclipse (`plugin.xml` e `build.properties`) as propriedades e labels para o nosso plug-in. O `plugin.xml` contém a informação das labels para os menus e

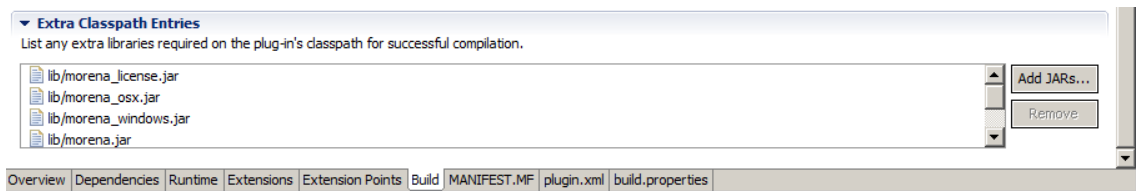
qual a classe que este menu deverá chamar, assim como a definição do caminho para um ícone, para atalho rápido. Também é possível adicionar uma “tooltip” para cada opção do menu.

Exemplo do plugin.xml:

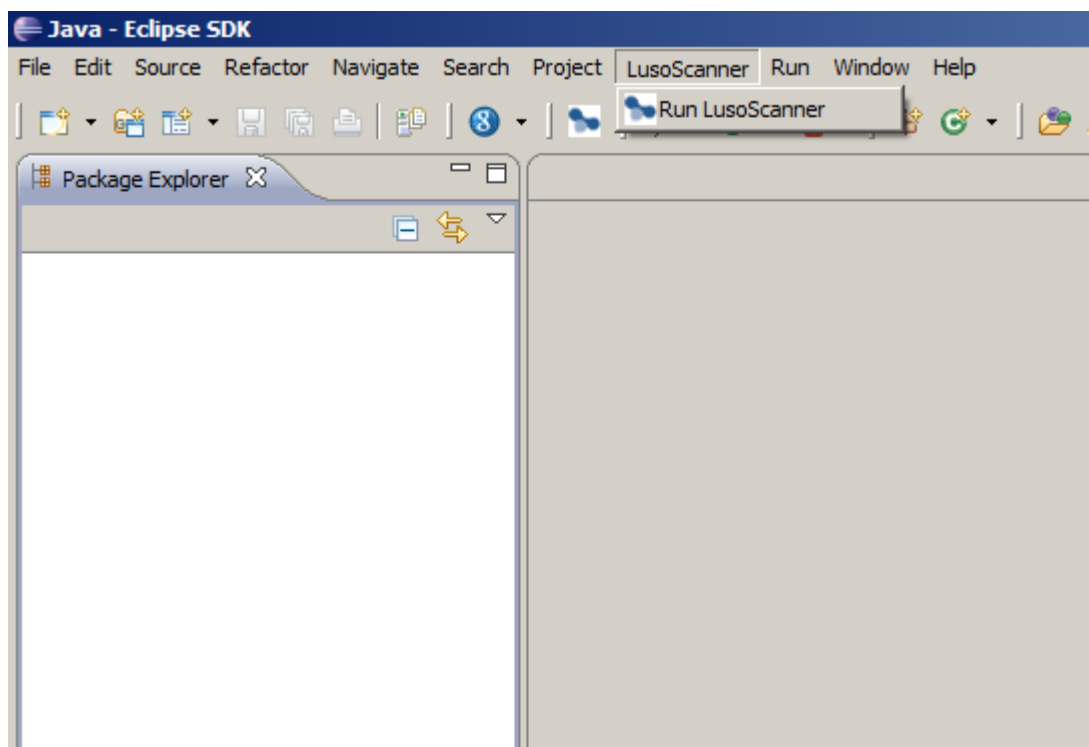
```
<extension
  point="org.eclipse.ui.actionSets">
  <actionSet
    label="LusoScanner"
    visible="true"
    id="com.example.hellonuno.actionSet">
    <menu
      label="LusoScanner"
      id="sampleMenu">
      <separator
        name="sampleGroup">
      </separator>
    </menu>
    <action
      class="com.example.lusodata.actions.OpenLusoScannerUIAction"
      icon="icons/lusoicon.png"
      id="com.example.hellonuno.actions.SampleAction"
      label="&Run LusoScanner"
      menubarPath="sampleMenu/sampleGroup"
      style="push"
      toolbarPath="sampleGroup"
      tooltip="LusoScanner">
    </action>
  </actionSet>
</extension>
```

Fig. 11 – Exemplo plugin.xml, com configurações da aplicação

No ficheiro build.properties é definido a classpath, ou seja, aqui é definido todas as livrarias necessárias para o funcionamento da aplicação. No meu caso, como utilizei a livraria Morena da Gnome e os JDBC's para as ligações às bases de dados, foram adicionadas as livrarias aqui (fig.12):



Após isto é possível correr o plug-in (fig.13):



Depois, foi criado o package `pt.lusodata.scanner`, com as classes (tal como o modelo de classes indica) `LusoScannerUI.java`, `LusoScanner.java`, `LusoScannerBD.java` e `LusoScannerECM.java` (que comunicam com a classe `LusoScanner`).

Começou então, a ser desenvolvido o UI da aplicação com as respectivas opções, na classe `LusoScannerUI`. A classe `LusoScanner` é que contém a lógica de toda a aplicação. As funções mais importantes desenvolvidas são:

<code>acquireImage();</code>	Capturar Imagem
<code>saveToFile();</code>	Guardar em ficheiro
<code>saveToBD();</code>	Guardar em Base de Dados

A função “`acquireImage`”, é a essencial para a captura de uma imagem de um dispositivo. Ela pergunta ao utilizador qual o driver que pretende utilizar (Twain ou SANE) e executa o seu método de captura. O código seguinte demonstra uma captura de imagem via livreria **Morena** (figura 14):

```
try
{
    status.setText("Working ...");
    MorenaSource source=Morena.selectSource(MainPanel.this);
    if (source!=null)
    {
        source.setColorMode();
        source.setResolution(100);
        while (true)
        {
            MorenaImage morenaImage=new MorenaImage(source);
            int imageStatus=morenaImage.getStatus();
            if (imageStatus==ImageConsumer.STATICIMAGEDONE)
            {
                int imageWidth=morenaImage.getWidth();
                int imageHeight=morenaImage.getHeight();
                int imagePixelSize=morenaImage.getPixelSize();
                ImagePanel image=new ImagePanel(Toolkit.getDefaultToolkit().createImage(morenaImage));
                MainPanel.this.add(image);
                select(image);
                int size=(int)Math.round(Math.sqrt(getComponentCount()));
                setLayout(new GridLayout(size, size));
                status.setText("Done - actual image size is "+imageWidth+" x "+imageHeight+" x "+imagePixelSize+" ...");
                validate();
                if (TwainSource.class.isInstance(source) && ((TwainSource)source).hasMoreImages())
                    continue;
            }
            else if (imageStatus==ImageConsumer.IMAGEABORTED)
                status.setText("Aborted, try again ...");
            else if (imageStatus==ImageConsumer.IMAGEERROR)
                status.setText("Failed, try again ...");
            break;
        }
    }
    else
        status.setText("Failed, try again ...");
}
```

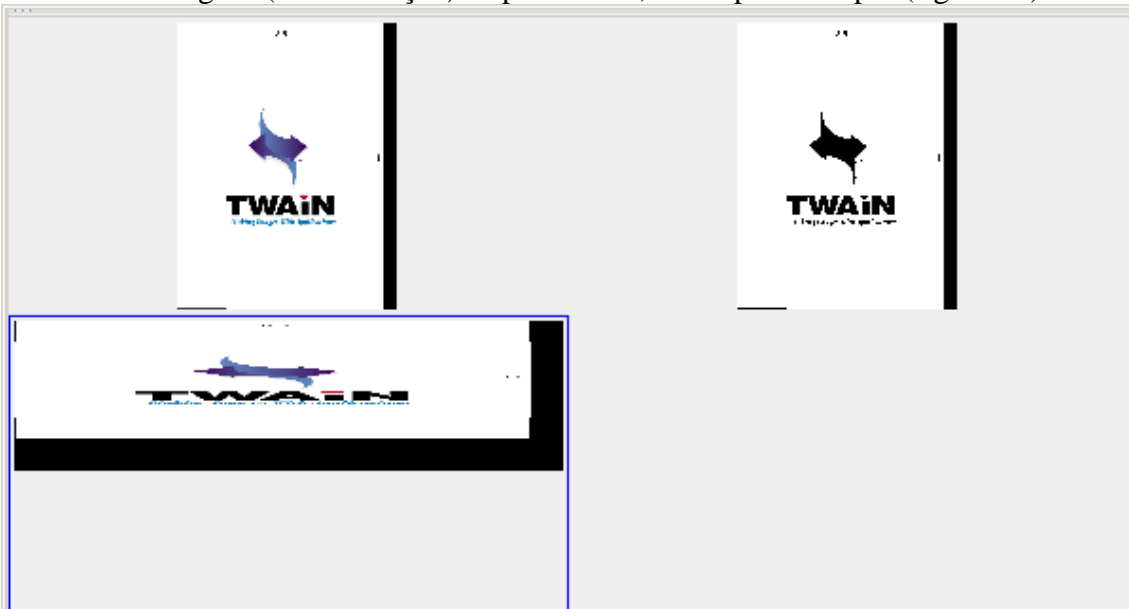
Neste caso, é utilizado o modo em cores, definido previamente pelo utilizador. Caso fosse pretendido o modo escalável de cinzentos, seria utilizado no código: `source.setGreyScaleMode()`. Basicamente a linha de código principal de para escolha de captura é:

```
MorenaSource source=Morena.selectSource(MainPanel.this);
```

E depois a imagem capturada, é criada em Java e adicionada à lista de imagens capturadas:

```
ImagePanel image=new  
ImagePanel(Toolkit.getDefaultToolkit().createImage(morenaImage))  
;MainPanel.this.add(image);
```

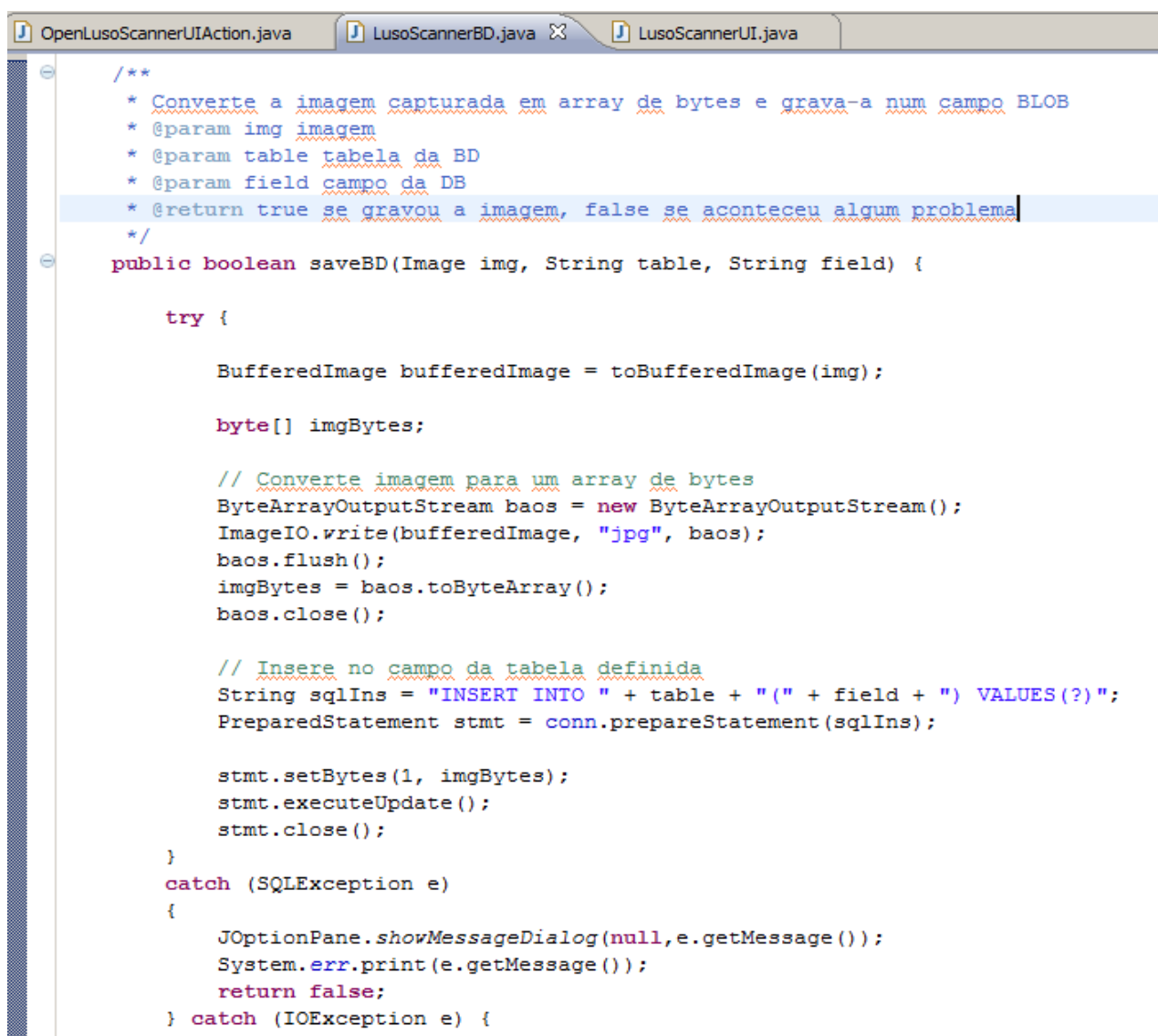
E a lista de imagens (com selecção) é apresentada, como por exemplo (figura 15):



A função “saveToFile”, trata da gravação da imagem seleccionada em ficheiro no sistema. Quando é pressionada esta opção, é aberta uma janela que pergunta o destino de salvaguarda da imagem seleccionada, com vários formatos de imagem suportados (jpeg, png, gif, etc), e o nome para o ficheiro de imagem. Este é o código da função (figura 16):

```
try
{
    status.setText("Working ...");
    Image image=selected.getImage();
    BufferedImage bufferedImage=new BufferedImage(image.getWidth(null), image.getHeight(null), BufferedImage.TYPE_INT_RGB);
    bufferedImage.createGraphics().drawImage(image, 0, 0, null);
    JFileChooser chooser=new JFileChooser();
    String e[] = ImageIO.getWriterFormatNames();
    for (int i=0; i<e.length; i++)
        chooser.addChoosableFileFilter(new Filter(e[i]));
    int result=chooser.showSaveDialog(MainPanel.this);
    if (result==JFileChooser.APPROVE_OPTION)
    {
        String ext=chooser.getFileFilter().getDescription();
        ext=ext.substring(0, ext.indexOf(' ')).toLowerCase();
        File file=chooser.getSelectedFile();
        String name=file.getName();
        if (!name.endsWith(ext))
            file=new File(file.getParentFile(), name+"."+ext);
        ImageIO.write(bufferedImage, ext, file);
    }
}
catch (Throwable exception)
{
    JOptionPane.showMessageDialog(MainPanel.this, exception.toString(), "Error", JOptionPane.ERROR_MESSAGE);
    exception.printStackTrace();
    status.setText("Failed, try again ...");
}
```

A função “saveToBD”, invoca a função saveBD() da classe LusoScannerBD. Esta função contém como parâmetros de entrada a imagem seleccionada, a nome da tabela e o campo onde o utilizador deseja guardar. Isto é tudo obtido na altura que o utilizador escolhe a opção de gravação em base de dados, são pedidos como entrada, o tipo de base de dados (MySQL ou SQLServer), o “hostname” do servidor de base de dados e o nome da base de dados a utilizar, o utilizador e password de acesso SQL, o nome da tabela e o nome do campo. Depois é construído um objecto da classe LusoScannerBD, que trata da recepção dos parâmetros de ligação à base de dados e cria uma ligação. Após feita a ligação, é então invocada a função saveBD com os seus respectivos parâmetros para a inserção da imagem na base de dados (figura 17):



```
OpenLusoScannerUIAction.java | LusoScannerBD.java | LusoScannerUI.java
/**
 * Converte a imagem capturada em array de bytes e grava-a num campo BLOB
 * @param img imagem
 * @param table tabela da BD
 * @param field campo da DB
 * @return true se gravou a imagem, false se aconteceu algum problema
 */
public boolean saveBD(Image img, String table, String field) {

    try {

        BufferedImage bufferedImage = toBufferedImage(img);

        byte[] imgBytes;

        // Converte imagem para um array de bytes
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(bufferedImage, "jpg", baos);
        baos.flush();
        imgBytes = baos.toByteArray();
        baos.close();

        // Inserir no campo da tabela definida
        String sqlIns = "INSERT INTO " + table + "(" + field + ") VALUES (?)";
        PreparedStatement stmt = conn.prepareStatement(sqlIns);

        stmt.setBytes(1, imgBytes);
        stmt.executeUpdate();
        stmt.close();

    }
    catch (SQLException e)
    {
        JOptionPane.showMessageDialog(null, e.getMessage());
        System.err.print(e.getMessage());
        return false;
    }
    catch (IOException e) {
```

A imagem é convertida num array de bytes e é feito um “INSERT” em sql, na tabela e campo pretendidos.

3.6 Testes

O ambiente de testes consistiu numa aplicação em Eclipse com o plug-in inserido, um servidor de base de dados MySQL e um servidor de base de dados SQLServer. Em ambos os servidores de bases de dados foi criado uma base de dados com uma tabela de teste e um campo de imagem, com o tipo de array de bytes.

Estas funções, assim como a interface gráfica e outras opções como o tratamento de cores, foram todas testadas no driver do TWAIN, foi exportado o plug-in para ser importando numa aplicação em Eclipse à parte, correu-se depois a aplicação e foi executadas várias capturas de imagem. Algumas delas foram gravadas em ficheiro e outras em bases de dados, concluindo assim a funcionalidade destas funções.

3.7 Implantação

A aplicação não chegou a ser implantada em ambiente de produção (em cliente) pois nem todas as funcionalidades estão prontas e este cliente em causa, está actualmente com outro projecto a decorrer com a Lusodata. As funcionalidades inacabadas são a ligação a um sistema ECM (como por exemplo o IBM Filenet ou Alfresco) e upload da imagem capturada, e a salvaguarda da configuração de propriedades para a ligação a estes sistemas num ficheiro de configuração (de forma a não serem repetidos como input, cada vez que é feito um upload).

Capítulo 4 Conclusões

4.1 Sumário

Os objectivos e funções principais foram implementados, assim como a aplicação embutida em plug-in para eclipse.

As funcionalidades desenvolvidas foram as seguintes:

- opções de captura
- gravar imagem capturada em disco
- gravar imagem captura em bases de dados (2 tipos, MySQL e SQLServer)
- listagem múltipla de imagens

4.2 Comentário Crítico

Algumas das funcionalidades não foram desenvolvidas ou ficaram inacabadas, devido ao facto de eu trabalhar na Lusodata há algum tempo e não estar propriamente a estagiar mas estar envolvido em projectos, o que me fez tirar algum tempo e perder algum apoio a nível deste projecto, assim como a mudança interna de coordenador. Contudo penso que alguns dos principais objectivos foram cumpridos assim como a exploração da biblioteca/livraria Morena.

4.3 Trabalho Futuro

No futuro pretende-se adicionar as funcionalidades que ainda faltam implementar (que faziam parte do projecto) e adicionar outras novas, tais como:

- Suporte de leitura OCR;
- Suporte de leitura de código de barras;
- Gravação de imagens em servidores FTP;
- Desenvolvimento de uma API para ponte entre java e twain através de c++ e java nativos.

Seria também mais interessante produzir uma ponte própria de comunicação entre o java e o twain, nativamente através de c++, sendo esta hipótese para o futuro e assim livra-se da dependência da biblioteca Morena.

Bibliografia

- [1] TWAIN (Standard for image acquisition devices). <http://www.twain.org>
- [2] SANE (Scanner Access Now Easy). <http://www.sane-project.org>.
- [3] jTWAIN (a Java Twain Implementation), <http://asprise.com/product/jtwain/>
- [4] jSANE (a Java Sane Implementation), <http://asprise.com/product/jsane/>
- [5] Morena (Gnome Software - Morena 6), <http://www.gnome.sk/Twain/jtp.html>
- [6] MMS Computing (Java Twain/Sane),
<http://thorntonzone.com/manuals/Compression/Fax,%20IBM%20MMR/MMSC/mmsc/index.html> ou <http://www.mms-computing.co.uk/>
- [7] JavaJeff Twain Example, <http://today.java.net/pub/a/today/2004/11/18/twain.html>
- [8] Eclipse (IDE, PDE), <http://www.eclipse.org> , <http://www.eclipse.org/pde/>
- [9] Rational Unified Process, [Ivar Jacobson](#), [Grady Booch](#), and [James Rumbaugh](#) (1999). *The Unified Software Development Process*
- [10] Imagem RUP, <http://en.wikipedia.org/wiki/File:Development-iterative.gif>
- [11] Diagrama de Actividade, http://pt.wikipedia.org/wiki/Diagrama_de_atividade
- [12] Diagrama de Sequência,
http://pt.wikipedia.org/wiki/Diagrama_de_sequ%C3%Aancia
- [13] Lusodata, apresentação/passado,
<http://www.lusodata.pt/webluso/apresenta%C3%A7%C3%A3o.aspx>

Apêndice

Rational Unified Process [9]

Fase de Concepção

A fase de concepção contém os *workflows* necessários para que as partes interessadas (*stakeholders*) concordem com os objectivos, arquitectura e o planeamento do projecto. Se as partes interessadas tiverem bons conhecimentos, então, pouca análise será requerida. Caso contrário, uma análise maior será requerida. Nesta fase os requisitos essenciais do sistema são transformados em casos de uso. O objectivo não é fechar todos os requisitos, mas apenas aqueles necessários para se formar uma opinião. Esta fase é geralmente curta e serve para se definir se vale a pena continuar com o projecto e definir os riscos e o custo deste. Um protótipo pode ser feito para que o cliente possa aprovar.

Como cita o RUP, o ideal é que sejam feitas iterações, mas estas devem ser bem definidas quanto a sua quantidade e objectivos.

Uma vez que neste caso não existe um cliente final, os requisitos são definidos com base nos objectivos a atingir e nas funcionalidades que são essenciais para que o sistema corresponda aos objectivos.

Fase de Elaboração

A fase de elaboração será apenas para o projecto do sistema, buscando complementar o levantamento / documentação dos casos de uso, voltado para a arquitectura do sistema, revisa a modelagem do negócio para os projectos e inicia a versão do manual do usuário. Deve-se aceitar: Visão geral do produto (incremento + integração) está estável? ; O plano do projecto é de confiança? ; Custos são admissíveis?

Fase de Construção

Na fase de construção, começa o desenvolvimento físico do software, produção de códigos, testes alfa e beta.

Deve-se aceitar testes, e processos de testes estáveis, e se os códigos do sistema constituem "baseline".

Fase de Transição

Nesta fase ocorre a entrega ("deployment") do software, é realizado o plano de implantação e entrega, acompanhamento e qualidade do software. Produtos (releases, versões) devem ser entregues, e ocorrer a satisfação do cliente. Nesta fase também é realizada a capacitação dos utilizadores.

