

**DEPARTAMENTO DE INFORMÁTICA**

**Mestrado em Engenharia Informática**

Dissertação

Plataforma Integrada para tratamento Electrónico de Documentos

**Edgar Miguel Sárinhas Marino (nº 18540)**

**ORIENTADOR: LUÍS ARRIAGA DA CUNHA**  
**COORDENADOR EXTERNO: VÍCTOR PARREIRA**

## Agradecimentos

---

Em primeiro lugar gostaria de agradecer a toda a minha família e amigos, por todo o seu apoio e incentivos, especialmente aos meus pais e à minha namorada que sempre me ajudaram. Sem eles com certeza que esta oportunidade não poderia ter sido concretizada.

Gostaria também de agradecer ao professor Luís Arriaga da Cunha da Universidade de Évora, pela orientação e ajuda ao longo deste projecto, bem como todos os ensinamentos que me transmitiu ao longo destes anos.

Um agradecimento muito especial para o Consultor Vítor Parreira, pela forma como me recebeu e me ajudou incondicionalmente, mas também por me ter dado a oportunidade de poder realizar este projecto.

Gostaria também de expressar a minha gratidão a todos os meus colegas de trabalho pela fácil integração que me proporcionaram.

Agradeço a todos os meus colegas e professores que me acompanharam no meu percurso escolar e académico, com particular atenção para meus colegas e amigos Valter Carvalho, Francisco Matias, André Teixeira e João Perdigão pela sua ajuda e apoio nos momentos em que mais necessitei.



485 656



# **Plataforma Integrada para tratamento Electrónico de Documentos**

---

Desde a entrada de facturas numa organização até ao seu lançamento, é necessário percorrer um longo caminho. Este caminho torna-se mais longo e menos eficaz se for efectuado manualmente, existindo, por exemplo, grande probabilidade de perda de facturas ou atrasos no lançamento das mesmas, principalmente em caso de organizações com grande volume de facturação. Com o aparecimento das plataformas de captura de dados este processo torna-se mais rápido e eficaz, permitindo às organizações libertar os seus funcionários para outras tarefas, diminuindo a quantidade de papel nas secretárias. Estas plataformas utilizam tecnologias de OCR, ICR, OMR, etc., que permitem reconhecer a informação que consta nos documentos. Os dados capturados são verificados pelos funcionários e posteriormente encaminhados para um ERP onde os documentos são lançados e contabilizados.



# **Integrated Platform for Documents Electronic Treatment**

---

From the moment an invoice is received in a company until it is registered in its information system, a long way has to be done. These steps become even longer and less effective if they are done manually and there is a high probability of losing invoices or creating unwanted delays, especially in companies with a high volume of invoicing.

With the Capture Platforms this process becomes faster and more efficient, allowing people to concentrate on other important tasks and making it possible to evolve to a paperless company.

These platforms use OCR, ICR OMR technologies, among others, that allow for the automatic capture of the document's content. The captured data are checked by the employees and forwarded to an ERP system where the documents are registered and processed.



# Índice Geral

---

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto Académico	2
1.2	Contexto Institucional	2
1.3	Contexto Tecnológico e Científico	2
1.4	Objectivos da Dissertação	3
1.5	Estrutura do Documento	3
<b>2</b>	<b>Soluções de Captura de Dados</b>	<b>4</b>
2.1	Enquadramento Histórico e Motivação	5
2.2	Definições de Tecnologias de Reconhecimento	7
2.3	Readsoft	8
2.3.1	Readsoft Documents For Invoices	8
2.3.2	Readsoft XML Input	13
2.3.3	Invoice Cockpit	16
2.3.4	Web Cycle	17
2.3.5	Web Board	17
2.3.6	Analyzer	18
2.3.7	MailRooms	18
2.3.8	Readsoft Documents For Forms	19
<b>3</b>	<b>ERP</b>	<b>24</b>
3.1	O que são ERP's	25
3.2	SAP	26
<b>4</b>	<b>Caso Prático</b>	<b>28</b>
4.1	Nota introdutória	29
4.2	Arquitectura	29
4.3	Instalação e Desenvolvimento	30
4.3.1	Análise e levantamento de necessidades	30

4.3.2	Separação dos documentos e arquivo físico	30
4.3.3	Instalação do Software Readsoft Documents For Invoices	31
4.3.4	Parametrização dos ficheiros de configuração da aplicação	31
4.3.5	Parametrização da aplicação	32
4.3.6	Criação de Nova coluna em Verify	33
4.3.7	Desenvolvimento de uma função que valida se o fornecedor existe para os ficheiros do tipo XML	34
4.3.8	Desenvolvimento em Visual Basic 6 de validações adicionais em Verify (calculo de tolerância)	37
4.3.9	Aplicação de comparação de versões de facturas	41
4.3.10	Desenvolvimento de plugin para transformação de ficheiros Tiff em Multi-tiff e PDF	51
4.3.11	Parametrização do ficheiro eicc.ini para enviar os dados para Cockpit/SAP	54
4.3.12	Parametrização de Cockpit	55
5	Conclusão	59
3.1	Apreciação Crítica/Avaliação	60
3.2	Conclusões Finais	61
6	Bibliografia	63

# Índice de Figuras

---

FIGURA 1 - PROPRIEDADES DE UM CAMPO DE CABEÇALHO.....	8
FIGURA 2 - MÓDULO MANAGER - INVOICES .....	9
FIGURA 3 - MÓDULO OPTIMIZE - INVOICES .....	10
FIGURA 4 - MÓDULO SCAN - INVOICES .....	11
FIGURA 5 - MÓDULO INTERPRET - INVOICES .....	12
FIGURA 6 - MÓDULO VERIFY - INVOICES.....	13
FIGURA 7 – FICHEIRO XML ORIGINAL.....	14
FIGURA 8 – FICHEIRO XML APÓS TRANSFORMAÇÃO .....	14
FIGURA 9 – FICHEIRO XSL .....	15
FIGURA 10 – FICHEIRO XSD.....	16
FIGURA 11 – ANALYZER.....	18
FIGURA 12 – MÓDULO MANAGER - FORMS .....	20
FIGURA 13 – MÓDULO SCAN - FORMS .....	20
FIGURA 14 – MÓDULO INTERPRET - FORMS .....	21
FIGURA 15 – MÓDULO VERIFY - FORMS .....	22
FIGURA 16 – MÓDULO TRANSFER - FORMS.....	22
FIGURA 17 – ARQUITECTURA DE APLICAÇÃO READSFOFT INTEGRADA COM SAP.....	29
FIGURA 18 – COLUNA ADICIONAL EM VERIFY.....	33
FIGURA 19 – MAPEAMENTO DA COLUNA NO MANAGER.....	34
FIGURA 20 – JANELA DE LOGIN DA APLICAÇÃO “CONFIRMA_FACTURAS” .....	42
FIGURA 21 – JANELA DE VERIFICAÇÃO DE VERSÕES DA APLICAÇÃO “CONFIRMA_FACTURAS” .....	44
FIGURA 22 – MAPEAMENTO DE CAMPOS EM COCKPIT .....	56
FIGURA 23 – DETALHE DE UMA FACTURA EM COCKPIT .....	57

## Termos abreviados, Definições

---

<b>Termo</b>	<b>Descrição</b>
ABAP	<i>Advanced Business Application Programming</i>
ASCII	<i>American Standard Code for Information Interchange</i>
BD	<i>Base de dados</i>
EBCDIC	<i>Extended Binary Coded Decimal Interchange Code</i>
ERP	<i>Enterprise Resource Planning</i>
ICR	<i>Intelligent Character Recognition</i>
IWR	<i>Intelligent Word Recognition</i>
OCR	<i>Optical Character Recognition</i>
OMR	<i>Optical Mark Recognition</i>
SAP	<i>System, Applications and Products in Data Processing</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>
XSL	<i>eXtensible Stylesheet Language</i>

# 1 Introdução

---

1.1	Contexto Académico	2
1.2	Contexto Institucional	2
1.3	Contexto Tecnológico e Científico	2
1.4	Objectivos da Dissertação	3
1.5	Estrutura do Documento	3

## **1.1 CONTEXTO ACADÉMICO**

Este documento foi escrito no contexto da dissertação final de curso do Mestrado em Engenharia Informática da Universidade de Évora. Trata-se de uma unidade curricular de 48 ECTS (*European Credit Transfer and Accumulation System*).

Como base de realização desta dissertação foi efectuado um estágio curricular, sendo a carga horária semanal média de 30 horas.

O estágio decorreu durante o período de 6 de Agosto de 2008 a 6 de Fevereiro de 2009.

A orientação pedagógica deste projecto esteve a cargo do Professor Luís Arriaga da Cunha do Departamento de Informática da Universidade de Évora.

O estágio curricular decorreu na instituição IDW (*Integration, Development and Warehousing*). A coordenação externa esteve a cargo de Victor Parreira.

## **1.2 CONTEXTO INSTITUCIONAL**

A instituição que me recebeu e possibilitou a realização deste projecto foi a IDW. Fundada em 2003, a IDW é uma empresa Portuguesa com apostas claras e inequívocas. Uma arquitectura de serviços e produtos coerente e consistente, que privilegia essencialmente a integração, as capacidades técnicas dos produtos e a capacidade de suporte e manutenção que pode ser oferecida. As principais áreas de actuação da empresa são: *Business Intelligence*, *Segurança*, *Consultoria*, *Business Process Automation*, *Data Management & Business Continuity* (IDW, 2003).

## **1.3 CONTEXTO TECNOLÓGICO E CIENTÍFICO**

Conseguir efectuar a contabilização e lançamento de documentos manualmente, tornava-se numa missão demorada e difícil. Com as aplicações de captura de dados associadas a aplicações de *Workflow* de aprovação e lançamento de documentos, este processo torna-se substancialmente mais

simples. Estas aplicações permitem uma automação de todo o circuito de vida dos documentos desde a entrada até à sua saída das organizações.

## **1.4 OBJECTIVOS DA DISSERTAÇÃO**

Esta dissertação tem por objectivo em primeiro lugar, conceituar e enquadrar historicamente os sistemas de desmaterialização de papel bem como abordar as suas características e funcionalidades. Pretende também mostrar a importância destes sistemas para empresas cujo volume de facturas é considerável. É também objectivo desta dissertação mostrar na prática como estes sistemas funcionam, através da integração do software *Readsoft Document For Invoices* com o ERP *SAP*.

## **1.5 ESTRUTURA DO DOCUMENTO**

A estrutura desta dissertação está organizada pelos seguintes capítulos:

**Soluções de Captura de Dados** – Este capítulo pretende enquadrar historicamente os sistemas de captura de dados e fornecer definições para os diferentes tipos de tecnologias de reconhecimento. Pretende também descrever as diferentes ferramentas da marca *Readsoft*.

**ERP** – Este capítulo descreve na generalidade o que são ERP's e mais especificamente quais as principais características do sistema *SAP*.

**Caso Prático** – Este capítulo, pretende demonstrar em termos práticos o funcionamento da aplicação *Readsoft Documents for Invoices* em conjunto com os desenvolvimentos específicos que foram efectuados para determinados módulos e a sua integração com o ERP *SAP (Cockpit)*.

**Conclusões** – Neste capítulo é feita uma apreciação crítica ao trabalho desenvolvido, são apresentados resumidamente os principais projectos desenvolvidos no âmbito desta dissertação, e por fim são apresentadas as conclusões.

# 2 Soluções de Captura de Dados

---

2	Soluções de Captura de Dados	4
2.1	Enquadramento Histórico e Motivação	5
2.2	Definições de Tecnologias de Reconhecimento	7
2.3	Readsoft	8
2.3.1	Readsoft Documents For Invoices	8
2.3.2	Readsoft XML Input	13
2.3.3	Invoice Cockpit	16
2.3.4	Web Cycle	17
2.3.5	Web Board	17
2.3.6	Analyzer	18
2.3.7	MailRooms	18
2.3.8	Readsoft Documents For Forms	19

## 2.1 ENQUADRAMENTO HISTÓRICO E MOTIVAÇÃO

Em 1929, Gustav Tauschek obteve a patente em OCR na Alemanha, seguido por Handel, que obteve uma patente norte americana em 1933. Depois disso, em 1935 Tauschek também conseguiu uma patente nos EUA com seu método.

A máquina de Tauschek era um dispositivo mecânico que usava modelos pré-definidos. Um sensor de luminosidade foi colocado de tal forma que quando um modelo e o caractere a ser reconhecido se alinhavam correctamente, a luz era direccionada completamente para eles e nenhuma luz era captada pelo sensor de luminosidade.

Em 1950, David H. Shepart, um especialista em criptografia das Forças Armadas da Agência de Segurança dos Estados Unidos, colaborou com o Dr. Louis Tordella na pesquisa de um procedimento para a automação de dados. Entre esses procedimentos, havia a demanda para conversão de mensagens impressas para linguagem de máquina. Shepard acreditava na possibilidade de criação de uma máquina capaz de fazer tal façanha. Shepard e um amigo, Harvey Cook, criaram então o "Gismo". Após isso, Sheppard fundou a Intelligent Machines Research Corporation (IMR), que depois começou a mostrar ao mundo as primeiras máquinas de OCR comercialmente.

Em 1965 a Readers Digest e a RCA colaboraram para construir um OCR leitor de documentos com o intuito de digitalizar números seriais de cupões retornados de anúncios. O processamento de documentos chegava a uma taxa de aproximadamente 1500 documentos por minuto.

Desde 1965 que o Serviço de Correios dos Estados Unidos tem utilizado máquinas OCR com o intuito de separar e ordenar correspondências. Instalações semelhantes foram também efectuadas na Europa, Inglaterra e Canadá nesse período. Posteriormente foi utilizado em conjunto com um sintetizador de voz para ajudar pessoas com deficiências visuais.

Com o decorrer dos anos, a aposta neste tipo de tecnologia foi crescendo gradualmente e começaram a surgir mais empresas a comercializar ferramentas baseadas neste tipo de tecnologia. A *Readsoft* foi uma dessas empresas e que surgiu em 1988 quando dois colegas suecos, Jan Andersson e Lars Appelstål, começaram a investir o tempo excedente deles no

desenvolvimento de um sistema para interpretar caracteres manuscritos. Este sistema teve o nome de Eyes & Hands.

O trabalho duro deles foi pago em 1992 quando a primeira licença de software foi vendida à firma de vendas por correspondência Yves Rocher.

Vindos da pequena companhia sueca com um revolucionário produto começaram sua expansão, Lars e Jan receberam cotações do mundo inteiro.

A companhia prosperou, e em 1994 Lars e Jan foram na primeira viagem de negócios transcontinental deles para visitar a feira AIIM nos Estados Unidos.

Aqui eles olharam brevemente o potencial enorme do mercado americano, e subsequentemente abriram uma subsidiária americana.

Em 1995, foram feitos planos novos para um produto totalmente novo nomeado *Invoices*. A ideia de um produto que automaticamente acha informação sobre uma factura sem campos pré-definidos era, e ainda é, revolucionário.

Em 1997 e 1998 eram anos de crescimento incrível para a *ReadSoft*. Subsidiárias eram estabelecidas em Espanha, França, e Austrália.

Em 1999, a *Readsoft* foi a empresa que mais cresceu na Suécia, este crescimento permitiu-lhe a entrada na bolsa de valores de Estocolmo. Em Outubro de 1999 a empresa chegou aos 2,000 accionistas, e a *ReadSoft* mostrou ser um dos verdadeiros vencedores daquele ano na bolsa de valores de Estocolmo.

Em 2001 o INVOICES teve um impacto verdadeiramente global, e as relações de parceria com os fornecedores dos principais sistemas de ERP como *Intentia*, *IFS* e *SAP* foram muito valorizadas.

Actualmente a *Readsoft* tem 12 subsidiárias em todo o mundo e mais de 3000 clientes em 70 países.

## 2.2 DEFINIÇÕES DE TECNOLOGIAS DE RECONHECIMENTO

Existem vários conceitos relacionados com as aplicações de extracção e captura de dados. Estas aplicações utilizam diversos tipos de tecnologia:

**OCR** - O reconhecimento óptico de caracteres (*OCR, optical character recognition*) converte imagens de texto como, por exemplo, documentos digitalizados, em caracteres de texto actuais. Também denominado reconhecimento de texto, o OCR permite editar e utilizar várias vezes o texto que normalmente está protegido dentro de imagens digitalizadas. O OCR funciona como um sistema de inteligência artificial, conhecido como reconhecimento de padrões, para identificar caracteres de texto individuais numa página, incluindo marcas de pontuação, espaços e fins de linhas. O OCR faz também reconhecimento de textos manuscritos, no entanto, a taxa de reconhecimento é muito inferior. O *Readsoft Documents For Invoices*, faz o reconhecimento dos dados através de OCR.

**OMR** - O Reconhecimento óptico de marcas (*OMR, Optical Mark Recognition*) é uma tecnologia semelhante ao OCR no entanto mais direccionada para o reconhecimento e extracção de informação de campos de marcações (*checkboxes*).

**IWR** - O Reconhecimento inteligente de palavras (*IWR, Intelligent Word Recognition*) é uma tecnologia que faz o reconhecimento de palavras impressas e manuscritas através de um dicionário definido pelo utilizador.

**ICR** - O reconhecimento inteligente de caracteres (*ICR, Intelligent Character Recognition*) tecnologia utilizada para identificar caracteres a partir de imagens digitalizadas aplicado em caracteres gerados de forma manuscrita ou impressa, com a capacidade de transformá-los em informação compreendida pelo computador, ou seja: Transformar imagem em caracteres ASCII, EBCDIC

ou HEXA. Esta tecnologia é utilizada pelo software Readsoft Documents For Forms.

## 2.3 READSOFT

### 2.3.1. READSOFT DOCUMENTS FOR INVOICES

O software *Readsoft Documents For Invoices* é utilizado para a captura de dados de documentos semi-estruturados, nos quais sabe-se qual o contexto da informação que se pretende, no entanto, não se conhece a estrutura do documento. Estes documentos são por exemplo:

- Facturas;
- Guias de Remessa;
- Pedidos de compra;

Este software utiliza a tecnologia de OCR para captar a informação pretendida. Esta informação é procurada por todo o documento e através de “titles” previamente parametrizadas, tal como é mostrado na figura:

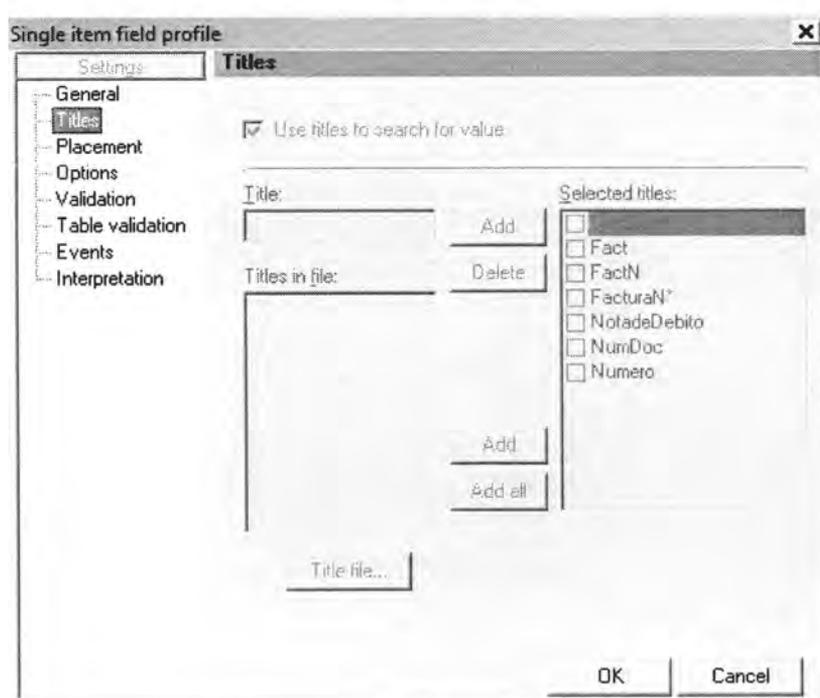


Figura 1 – Propriedades de um campo de cabeçalho.

Como mostra a figura em cima, o software vai procurar em todo o documento os dados pretendidos através dos “titles” definidos. Esta aplicação é muito flexível, porque é muito ajustável às necessidades das diferentes empresas, permitindo não só vários tipos de parametrizações como também embutir código nos diferentes módulos da aplicação. A aplicação é dividida em 6 módulos:

- **Manager**

Este é um módulo de administração, no qual se efectuam as parametrizações da aplicação. Neste módulo definem-se perfis, utilizadores, parametrizam-se as definições para os restantes módulos (*Scan, Interpret, Verify, Transfer*) etc. Este módulo é normalmente, apenas utilizado pelo administrador da aplicação.

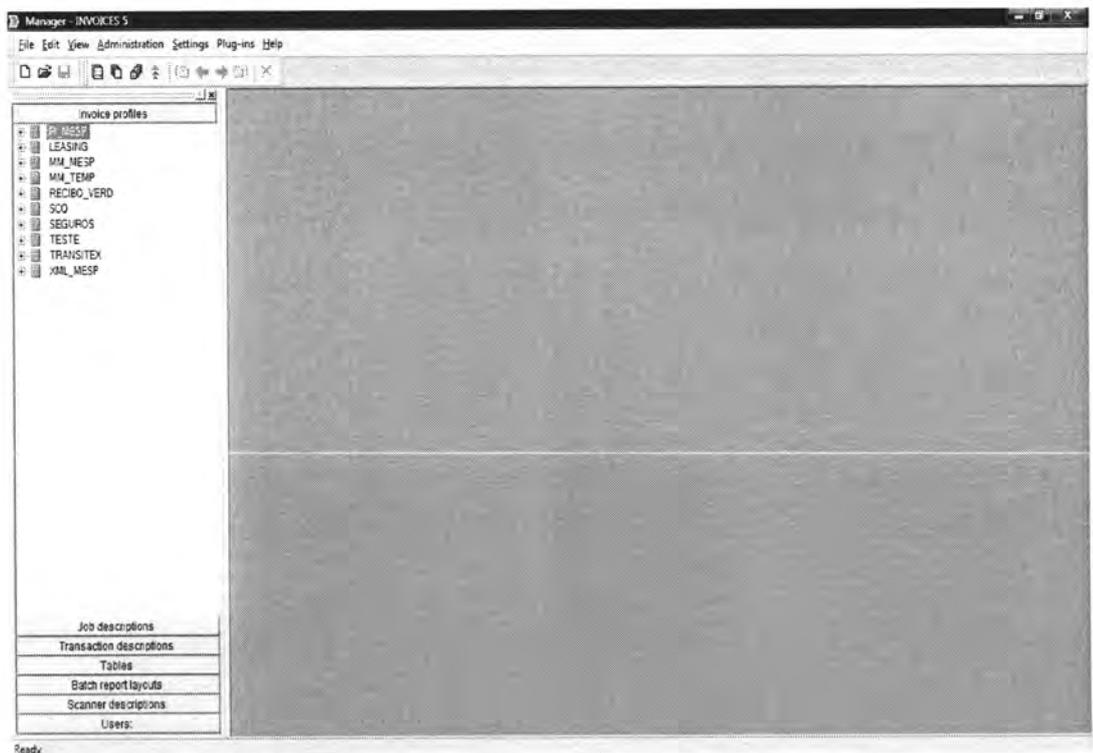


Figura 2 – Módulo Manager - Invoices.

- **Optimize**

Módulo também de administração que permite otimizar algumas parametrizações definidas no módulo *Manager*. Neste módulo é possível melhorar as definições por fornecedor e tipo de documento. Pode-se otimizar a informação a captar campo a campo, especificando exactamente a zona de captura que se pretende, ou por exemplo, indicar à aplicação que para aquele tipo de documento determinados campos não interessam captar.

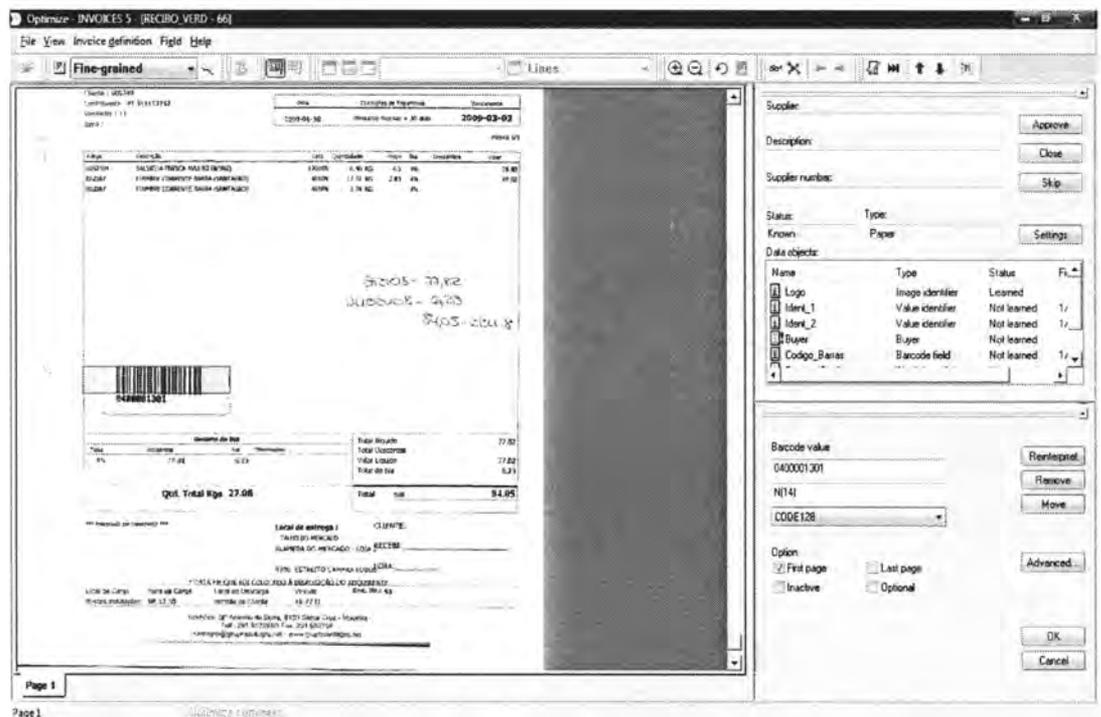


Figura 3 – Módulo Optimize - Invoices.

- **Scan**

O processo de captura de dados começa neste módulo. O utilizador coloca as facturas no alimentador de papel do Scanner para que estas sejam digitalizadas e separadas consoante for o método de separação definido. Existem três possibilidades de separação de documentos:

- Código de barras – A aplicação detecta o código de barras na primeira página do documento e considera que todas as páginas seguintes sem código de barras pertencem ao mesmo documento;

- Folha em branco – A aplicação considera que até encontrar uma página em branco todas as folhas digitalizadas pertencem ao mesmo documento.
- Página única – Se os documentos a serem tratados sejam de uma única página é possível parametrizar a aplicação desta forma.

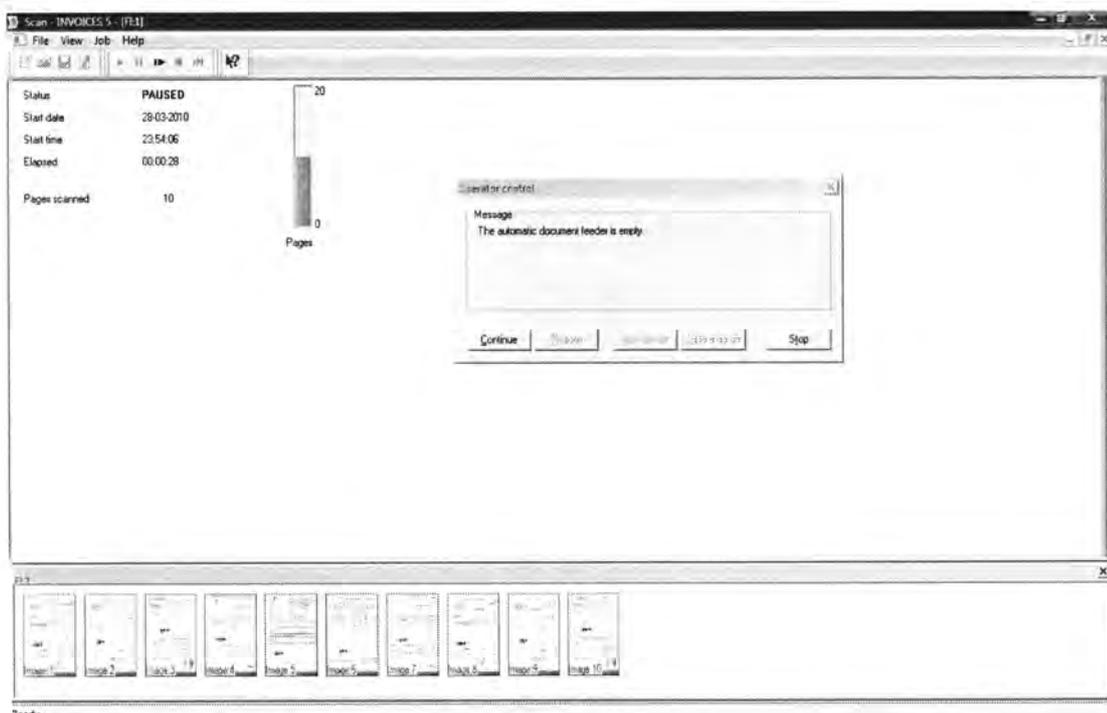


Figura 4 – Módulo Scan - Invoices.

- **Interpret**

O OCR é efectuado neste módulo. Aqui a aplicação vai buscar as imagens geradas pelo módulo *Scan*, ou imagens de uma determinada directoria e percorre o documento para procurar os “titles” definidos e através deles encontrar e interpretar os valores a capturar. Este módulo não tem interacção com o utilizador final dado que pode correr como serviço Windows. Caso não se configure como serviço Windows, este módulo pode ser executado manualmente tendo o seguinte aspecto:

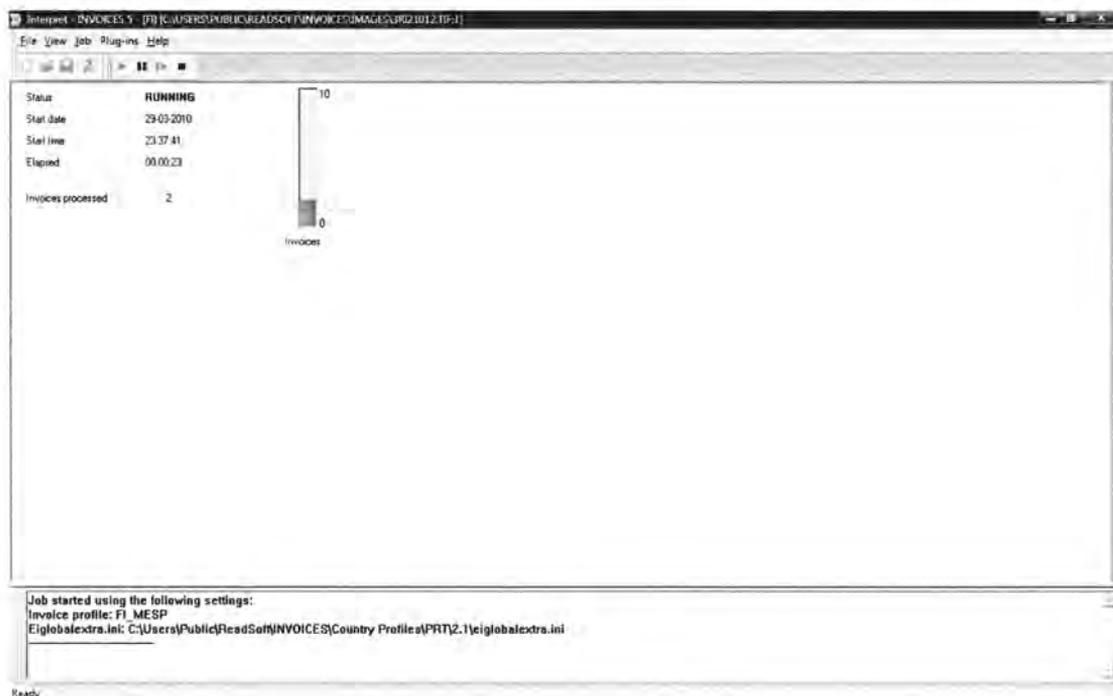


Figura 5 - Módulo Interpret - Invoices.

- **Verify**

Este é o módulo com mais interacção com o utilizador, no qual é efectuado o controlo de qualidade dos dados capturados pelo módulo *Interpret*. Aqui o utilizador pode validar todos os dados e corrigir os que estejam errados, pode rejeitar documentos, reenviá-los para nova interpretação, associá-los a outro perfil. A partir do momento em que o utilizador confirma que está tudo ok, não é mais possível alterar qualquer informação nesse documento, só o administrador do software o pode eliminar da base de dados. Dada a importância deste módulo é possível configurar, por exemplo, que cada utilizador receba apenas documentos associados a determinadas empresas e dessa forma cada utilizador só consegue visualizar os documentos que lhe estão destinados.

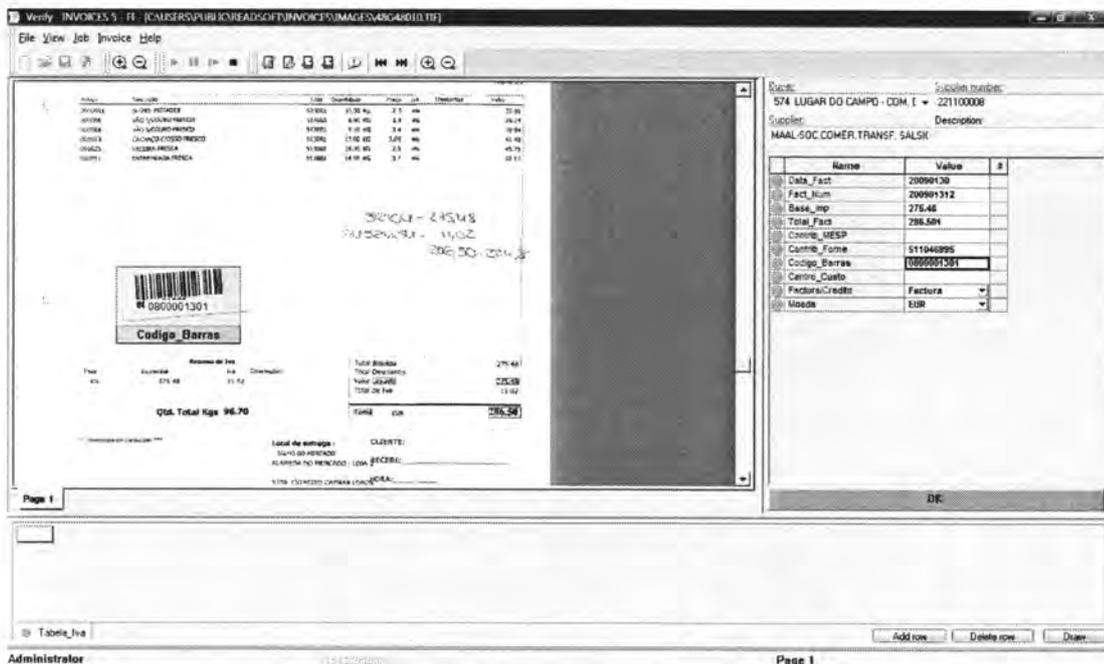


Figura 6 – Módulo Verify - Invoices.

- **Transfer**

Este módulo tal como o módulo *Interpret* pode correr como serviço Windows, pelo que também não tem interacção com o utilizador. Neste módulo os dados que foram verificados e aceites em *Verify* são enviados para um ficheiro de output previamente definido (.txt,.csv,.xls, etc), para uma base de dados ou então enviados para um ERP (exemplo: SAP), no qual estes documentos serão lançados contabilisticamente. Tal como o módulo *Interpret*, o *Transfer* também pode ser executado manualmente.

### 2.3.2. READSOFT XML INPUT

O *Readsoft Documents for Invoices* para além de tratar documentos em papel, também trata documentos electrónicos em diferentes formatos, como por exemplo o XML. Para tratar este tipo de documentos utiliza-se a aplicação XML Input que recebe o ficheiro XML proveniente de uma aplicação externa e o converte no formato legível pelo *Readsoft Documents for Invoices*. A conversão é efectuada através da utilização de ficheiros XSL e XSD.

## O XML Input pega no ficheiro em formato XML original:

```

<?xml version="1.0" encoding="utf-8" ?>
<Invoice>
  <SellerPartyDetails>
    <SellerOrganisationName>Mota Engil Eng. e Constr.</SellerOrganisationName>
    <SellerOrganisationTaxCode>PT500197814</SellerOrganisationTaxCode>
  </SellerPartyDetails>
  <BuyerPartyDetails>
    <BuyerOrganisationName>MESP-MOTA ENGIL S.PART.ADM.GESTAO</BuyerOrganisationName>
    <BuyerOrganisationTaxCode>PT505936364</BuyerOrganisationTaxCode>
  </BuyerPartyDetails>
  <InvoiceDetails>
    <DOCID>200916021707430877375267478276_1000</DOCID>
    <LINK>http://EBC-CIC-PRD/ebcadminwebpoint/RedirectInbound.aspx?ID=200916021707430877375267478276</LINK>
    <ATTACH>\\EBC-CIC-PRD\EFact\F_RECEB\ATTACHS\4d0da3ad-084b-4b7a-b22b-13951c4526c0.zip</ATTACH>
    <InvoiceTypeText>Factura</InvoiceTypeText>
    <InvoiceNumber>2200000064</InvoiceNumber>
    <InvoiceDate>20090212</InvoiceDate>
    <FICHEIRO>4d0da3ad-084b-4b7a-b22b-13951c4526c0.pdf</FICHEIRO>
    <DOCUMENTORIGINAL />
    <InvoiceTotalVatExcludedAmount>81,30</InvoiceTotalVatExcludedAmount>
    <InvoiceTotalVatIncludedAmount>81,30</InvoiceTotalVatIncludedAmount>
    <MOEDA>EUR</MOEDA>
    <QuantiaComIVAMoedaInterna>81,30</QuantiaComIVAMoedaInterna>
    <QuantiaSemIVAMoedaInterna>81,30</QuantiaSemIVAMoedaInterna>
    <TAXACAMBIO>1,00000</TAXACAMBIO>
    <CONDICAOPAGAMENTO>1500</CONDICAOPAGAMENTO>
    <SWIFT />
    <TIPODOCUMENTO>fact</TIPODOCUMENTO>
    <IBAN />
  </InvoiceDetails>
</Invoice>

```

Figura 7 – Ficheiro XML original.

E converte-o no ficheiro XML com uma estrutura compatível com a do *Readsoft Documents For Invoices*

```

<?xml version="1.0" encoding="utf-8" ?>
<Documents>
  <Document Version="1.1" Name="XML Invoice" Type="Invoice">
    <DocumentReferences>
      <DocumentReference>\\EBC-CIC-PRD\EFACT\F_RECEB\PDF\c2485102dbc74e6d8987d304ba3e0f7.org.xml</DocumentReference>
    </DocumentReferences>
    <ImageReferences>
      <ImageReference>4d0da3ad-084b-4b7a-b22b-13951c4526c0.pdf</ImageReference>
    </ImageReferences>
    <CertificateReference />
  </Invoice>
  <Supplier>
    <Name>Mota Engil Eng. e Constr.</Name>
    <SupplierVATRegistrationNumber>PT500197814</SupplierVATRegistrationNumber>
  </Supplier>
  <Buyer>
    <Name>MESP-MOTA ENGIL S.PART.ADM.GESTAO</Name>
    <BuyerVATRegistrationNumber>PT505936364</BuyerVATRegistrationNumber>
  </Buyer>
  <Fields>
    <Field Type="DOCID">200916021707430877375267478276_1000</Field>
    <Field Type="LINK">http://EBC-CIC-PRD/ebcadminwebpoint/RedirectInbound.aspx?ID=200916021707430877375267478276</Field>
    <Field Type="ATTACH">\\EBC-CIC-PRD\EFact\F_RECEB\ATTACHS\4d0da3ad-084b-4b7a-b22b-13951c4526c0.zip</Field>
    <Field Type="InvoiceNumber">2200000064</Field>
    <Field Type="InvoiceDate">20090212</Field>
    <Field Type="FICHEIRO">4d0da3ad-084b-4b7a-b22b-13951c4526c0.pdf</Field>
    <Field Type="DOCORIGINAL" />
    <Field Type="InvoiceTotalVatExcludedAmount">81,30</Field>
    <Field Type="InvoiceTotalVatAmount" />
    <Field Type="InvoiceTotalVatIncludedAmount">81,30</Field>
    <Field Type="MOEDA">EUR</Field>
    <Field Type="QuantiaComIVAMoedaInterna">81,30</Field>
    <Field Type="QuantiaSemIVAMoedaInterna">81,30</Field>
    <Field Type="TAXACAMBIO">1,00000</Field>
    <Field Type="CONDICAOPAGAMENTO">1500</Field>
    <Field Type="SWIFT" />
    <Field Type="TIPODOCUMENTO">fact</Field>
    <Field Type="IBAN" />
    <Field Type="SupplierVATRegistrationNumber">PT500197814</Field>
  </Fields>
</Documents>

```

Figura 8 – Ficheiro XML após transformação.

Para efectuar esta transformação são utilizados dois ficheiros:

- Ficheiro XSL que vai “traduzir” os campos do ficheiro XML original para o XML legível pelo *Invoices*, ou seja, é um vocabulário de XML criado para a finalidade exclusiva de transformar originais de XML de um estado a outro. Esse estado pode ser de XML a XML, de XML ao HTML, de XML ao texto, ou de XML a todo o outro formulário.

O XSL pode dividir-se em duas partes, uma para transformar o documento XML noutro tipo de documento (usando para isso o XSLT - linguagem de transformação) e uma outra parte para definir objectos de formatação para apresentação gráfica (XSLFO - linguagem de formatação de objectos).

```

- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" indent="yes" version="1.0" />
  <xsl:template match="/Invoice">
    <Documents>
      <Document Version="1.1" Name="XML Invoice" Type="Invoice">
        <DocumentReferences />
        <ImageReferences>
          <ImageReference>
            <xsl:value-of select="InvoiceDetails/FICHEIRO" />
          </ImageReference>
        </ImageReferences>
        <CertificateReference />
      </Document>
      <Invoice>
        <Supplier>
          <Name>
            <xsl:value-of select="SellerPartyDetails/SellerOrganisationName" />
          </Name>
          <SupplierVATRegistrationNumber>
            <xsl:value-of select="SellerPartyDetails/SellerOrganisationTaxCode" />
          </SupplierVATRegistrationNumber>
        </Supplier>
        <Buyer>
          <Name>
            <xsl:value-of select="BuyerPartyDetails/BuyerOrganisationName" />
          </Name>
          <BuyerVATRegistrationNumber>
            <xsl:value-of select="BuyerPartyDetails/BuyerOrganisationTaxCode" />
          </BuyerVATRegistrationNumber>
        </Buyer>
      </Invoice>
    </Documents>
  </template>
</stylesheet>

```

Figura 9 – Ficheiro XSL.

- XSD - XML Schema é uma linguagem baseada no formato XML para definição de regras de validação ("esquemas") em documentos no formato XML. A figura seguinte mostra um exemplo destes esquemas:

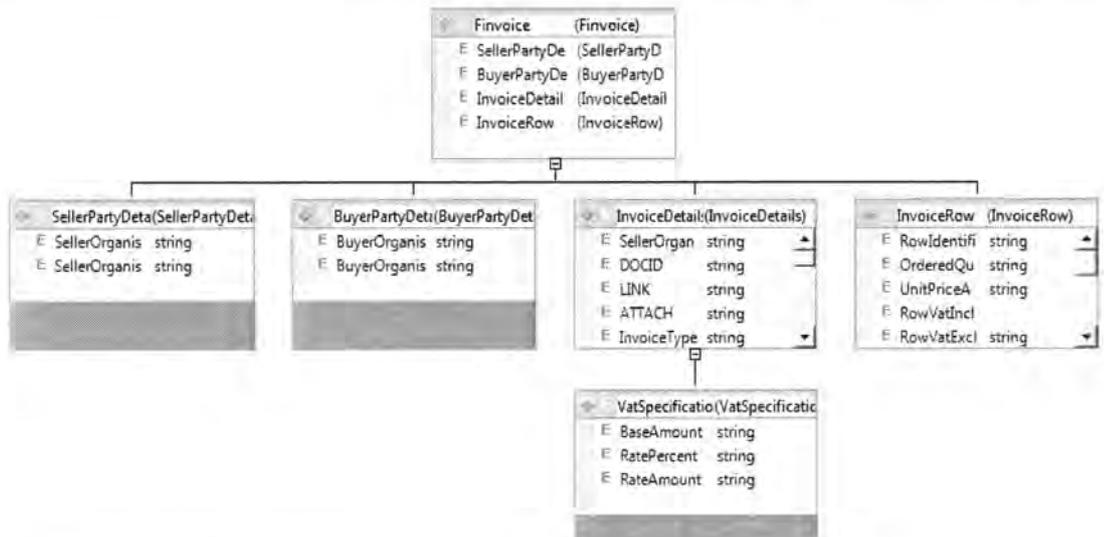


Figura 10 – Ficheiro XSD.

Após a transformação ser efectuada, o documento XML é movido para uma pasta onde o *Interpret* o vai buscar para interpretar e o colocar disponível para verificação.

### 2.3.3. INVOICE COCKPIT

Cockpit é um módulo desenvolvido pela *Readsoft* que faz a integração de documentos provenientes do *Invoices* no ERP SAP. Desta forma é possível processar dentro de *SAP* os documentos capturados pelo *Invoices*. Assim o processo de gestão de documentos fica completo. Das várias funcionalidades que acompanham este módulo podem-se se destacar:

- Transferência dos dados das facturas para o sistema *SAP*;
- Contabilização ou contabilização preliminar de facturas utilizando a tecnologia *SAP BAPI*;
- Apresentação de todos os dados dos documentos incluindo a sua imagem, com a possibilidade de os completar ou corrigir;

- Integração das transacções de *SAP* para permitir a visualização e gestão das ordens de compra;
- Conexão com os dados maestros de *SAP*;
- Gestão on-line dos documentos incompletos mediando o estabelecimento por defeito das transacções *SAP* quando se gerem e comprovam facturas;
- Permite a conexão ao *SAP Business Workflow*;
- Permite utilizar as ferramentas *standard* de *SAP* sem ter que efectuar qualquer alteração em *SAP*;
- Permite a contabilização periódica automática das facturas;

#### 2.3.4. WEB CYCLE

*Web Cycle* é uma extensão do *Invoice Cockpit*, que permite a automatização e optimização de processos. O *Web Cycle* pode ser considerado o *workflow* da *Readsoft*. As facturas circulam dentro da organização em formato electrónico e é possível colocar informação adicional e por outro lado o *Invoices Cockpit* garante um registo completo e preciso de todas as transacções, dando também a possibilidade de saber quem está a trabalhar em todo o momento em cada factura.

#### 2.3.5. WEB BOARD

O *Web Board* é uma aplicação Web que conecta com o cockpit e permite aos fornecedores da organização acederem externamente à organização e consultarem o estado das suas facturas. As principais funcionalidades são:

- Transmissão de dados e imagens das facturas via Web;
- Transparente: cada fornecedor só tem acesso aos seus pedidos;
- Os fornecedores podem consultar o estado das suas facturas a qualquer momento de qualquer lugar que tenha acesso à internet;

### 2.3.6. ANALYZER

*Analys*er é uma extensão de cockpit que permite a consulta de estatísticas de tempo de processamento de facturas, análise de mensagens de erro e verificar quais são mais comuns.

Report	Count (FI+MM)	Count (FI)	Count (MM)	Gross Amount
All documents	636	241	394	31.99
Status: Unprocessed	13	2	10	62
Status: Errors	400	143	256	12.80
Status: Posted	180	66	95	4.58
Status: Web Cycle	52	30	22	13.08
Status: Rejected	1	0	1	
Posting type: Online	138	63	75	1.49
Posting type: Submit	19	1	18	2
Posting type: Repetitor	0	0	0	
Posting type: Web Cycle	0	0	0	
Posting type: EDI	0	0	0	
Payment lock: Free for payment	583	241	322	28.94
Payment lock: Invoice verification (R)	72	0	72	2.15
Average lead time	18 d 21 h 21 m 32 s	26 d 14 h 33 m 47 s	13 d 11 h 42 m 57 s	
Average Web Cycle processing time	19 d 17 h 08 m 17 s	19 d 01 h 28 m 59 s	21 d 02 h 45 m 37 s	

Figura 11 – Analyser.

Podem-se destacar como principais funcionalidades:

- Análise detalhada da qualidade dos dados;
- Análise da taxa de automatização e gestão das facturas;
- Análise de quantidade de documentos por fornecedor;

### 2.3.7. MAILROOMS

O *MailRooms* é um software muito útil para indexar e classificar documentos. Este software separa todo o tipo de documentação que entra numa determinada organização (revistas, cartas, e-mails, facturas, formulários, etc.)

e separa-os consoante tipos de documento previamente parametrizados. Após ser efectuada esta classificação é efectuada a indexação documento a documento extraindo a informação pretendida do mesmo para depois esta informação ser integrada na plataforma de gestão documental da organização consoante for a classe documental a que o documento pertence.

### 2.3.8. READSOFT DOCUMENTS FOR FORMS

O software *Readsoft Documents For FORMS* é utilizado para a captura de dados de documentos estruturados (formulários), nos quais se conhece exactamente a estrutura do documento e o sítio a capturar, sem necessidade de saber o seu contexto.

Este software utiliza a tecnologia de ICR para captar a informação pretendida. Previamente parametriza-se um template no qual se definem os campos a capturar e a sua localização exacta. Desta forma a aplicação sabe exactamente onde buscar a informação. Tal como o *Readsoft Documents For Invoices*, esta aplicação também se divide em vários módulos:

- **Manager**

Este é um módulo de administração, no qual se efectuem as parametrizações da aplicação. Neste módulo definem-se *FormDefinitions*, utilizadores, parametrizam-se as definições para os restantes módulos, etc. Este módulo é normalmente, apenas utilizado pelo administrador da aplicação.

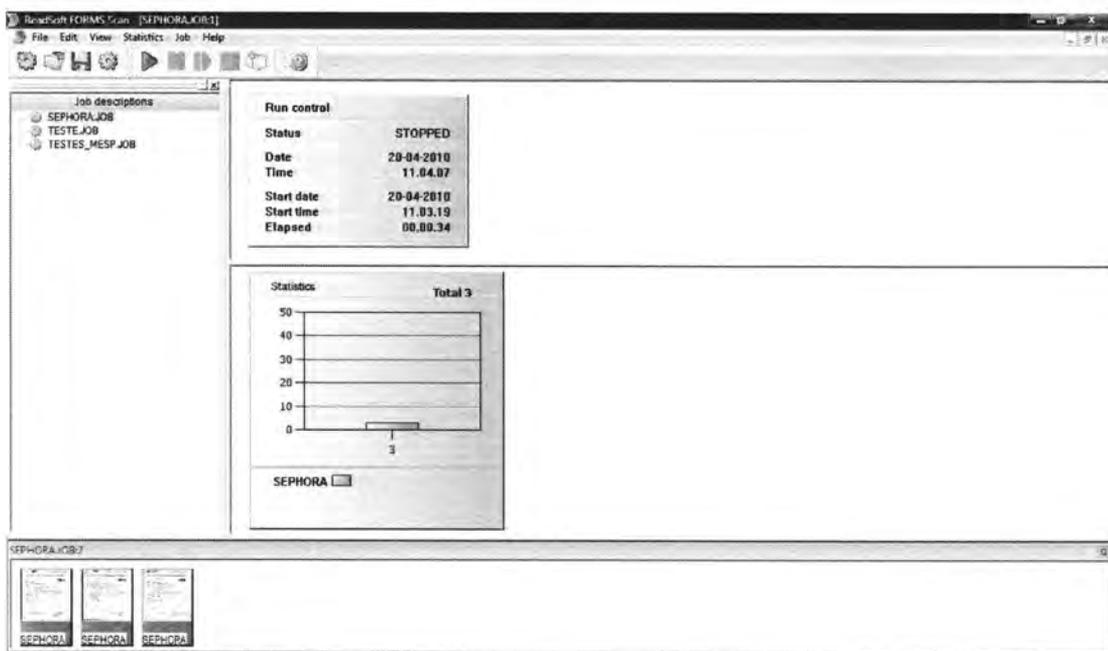


For Help, press F1

Figura 12 – Módulo Manager – Forms.

- Scan

O processo de captura de dados começa neste módulo. O utilizador coloca os formulários no alimentador de papel do Scanner para que estes sejam digitalizados e identificados segundo a *FormDefinition* (template) utilizada para digitalizar.



Ready.

Figura 13 – Módulo Scan – Forms.

- **Interpret**

O ICR é efectuado neste módulo. Aqui a aplicação vai buscar as imagens geradas pelo módulo de Scan, ou imagens de uma determinada directoria e interpreta campo a campo (previamente definidos) os caracteres manuscritos ou impressos do formulário. Este módulo não tem interacção com o utilizador final dado que pode correr como serviço Windows.

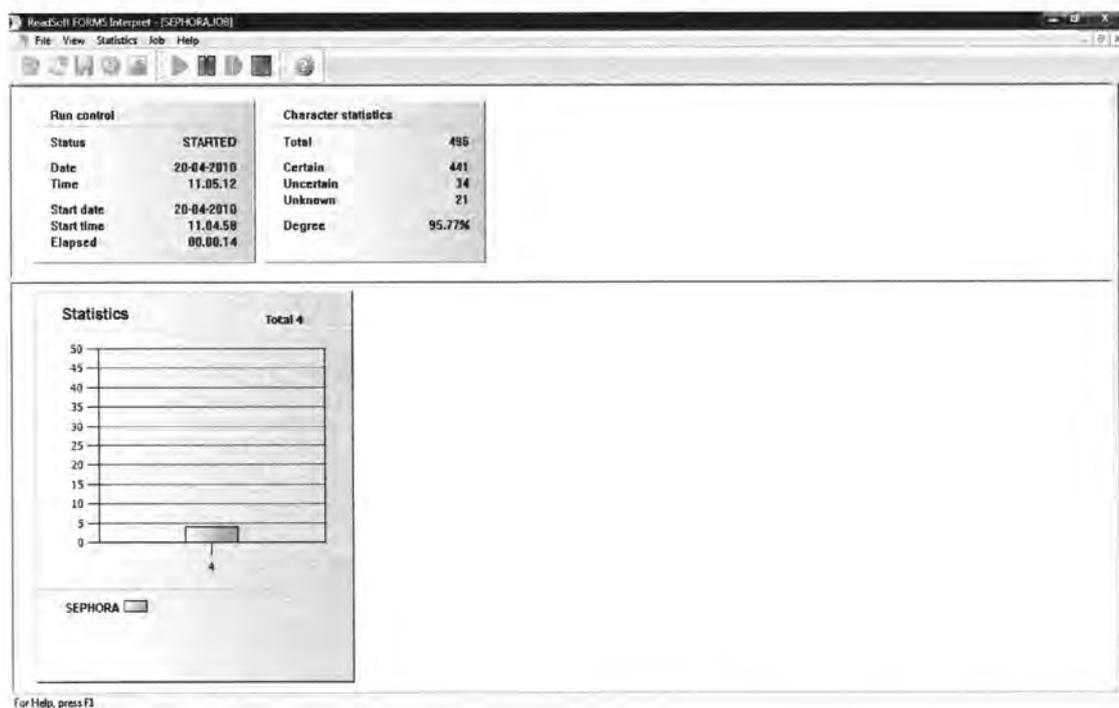


Figura 14 - Módulo Interpret - Forms.

- **Verify**

Este é o módulo com mais interacção com o utilizador, no qual é efectuado o controlo de qualidade dos dados capturados pelo módulo *Interpret*. Aqui o utilizador pode validar todos os dados e corrigir os que estejam errados, pode rejeitar documentos, apagá-los da base de dados ou simplesmente aprová-los. A partir do momento em que o utilizador confirma que está tudo ok, não é mais possível alterar qualquer informação nesse documento, só o administrador do software o pode eliminar da base de dados.

Figura 15 – Módulo Verify – Forms.

- **Transfer**

Este módulo tal como o módulo *Interpret* pode correr como serviço Windows, pelo que também não tem interação com o utilizador. Neste módulo os dados que foram verificados e aceites em *Verify* são enviados para um ficheiro de output previamente definido (.txt,.csv,.xls, etc).

Figura 16 – Módulo Transfer – Forms.

Tanto o módulo Interpret como o módulo Transfer do Forms podem também ser executados através de JobLists, que consistem em listas de JobDescription que são executadas sequencialmente.

# 3 ERP

---

3	ERP	24
3.1	O que são ERP's	25
3.2	SAP	26

### 3.1 O QUE SÃO ERP'S

“Os ERP's são sistemas de informação que integram todos os dados e processos de uma organização num único sistema. Esta integração está direccionada para:

- Finanças;
- Contabilidade;
- Recursos humanos;
- Fabricação;
- Marketing;
- Vendas;
- Compras;
- Processamento de transacções;
- Sistemas de informações gerenciais;
- Sistemas de apoio à decisão;

Os ERP's em termos gerais, são uma plataforma de software desenvolvida para integrar os diversos departamentos de uma empresa, possibilitando a automação e armazenamento de todas as informações de negócios.

Assim sendo os ERP's proporcionam às empresas diversas vantagens, tais como:

- Eliminar o uso de interfaces manuais;
- Reduzir custos;
- Optimizar o fluxo da informação e a qualidade da mesma dentro da organização (eficiência);
- Optimizar o processo de tomada de decisão;
- Eliminar a redundância de actividades;
- Reduzir os limites de tempo de resposta ao mercado;
- Reduzir as incertezas do *Lead time*;

- Incorporação de melhores práticas (codificadas no ERP) aos processos internos da empresa;”

In <http://wapedia.mobi/pt/ERP#1>.

Apesar deste conjunto de vantagens, a utilização do ERP por si só não torna os processos totalmente optimizados, é por isso, necessário a utilização de outras ferramentas para tornar os processos o mais optimizados possível.

### **3.2 SAP**

“A *SAP* é uma empresa alemã criadora do sistema integrado de gestão empresarial (ERP) *SAP ERP*. Este sistema procura contemplar a empresa como um todo, dividido em módulos:

- SAP MM - *Material Management*;
- SAP SD - *Sales and Distribution*;
- SAP FI - *Financial*;
- SAP PP - *Production Planing*;
- SAP HR - *Human Resources*;
- SAP PS - *Project System*;
- SAP CO - *Controlling*;
- SAP WF - *Work Flow*;
- SAP QM - *Quality Management*;
- SAP WM - *Warehouse Management*;
- SAP AM - *Fixed Assed Management*;
- SAP PM - *Plant Maintenance*;
- SAP IS - *Industry Solutions*;
- SAP BW - *Businesse Warehouse*;

Cada módulo corresponde a uma área específica. Cada programa, é executado através de uma transacção separadamente. Estes programas são desenvolvidos em ABAP, uma linguagem de programação, na qual a *SAP* detém os direitos e que a sintaxe é muito semelhante à do COBOL.

O primeiro produto importante da *SAP* foi R/2, que até então era um conjunto de módulos voltado para mainframes, até que em 1995 foram apresentadas as primeiras aplicações do R/3, que ao contrário do seu antecessor, era voltado para o ambiente cliente-servidor, sendo assim, não uma versão nova do R/2, mas sim um sistema diferente.

Um sistema *SAP* R/3 é composto por três camadas:

- *Frontend* - Camada responsável pela exibição dos ecrãs ao utilizador.
- *Application* - Onde são processadas as operações efectuadas, transferindo para o *Frontend*, os dados a serem exibidos. É nesta camada que os programas ABAP são executados.
- *Database* - O Dicionário de Dados no R/3 é uma abstracção de um (SGBD) Sistema de Gestão de Base de Dados, e permite uma gestão centralizada de todas as definições de dados. Nele podem-se criar todos os tipos de dados, para uso em programas ABAP ou em interfaces de módulos de função.”

In [http://wapedia.mobi/pt/SAP\\_ERP](http://wapedia.mobi/pt/SAP_ERP)

# 4

## Caso Prático

---

4	Caso Prático	28
4.1	Nota introdutória	29
4.2	Arquitectura	29
4.3	Instalação e Desenvolvimento	30
4.3.1	Análise e levantamento de necessidades	30
4.3.2	Separação dos documentos e arquivo físico	30
4.3.3	Instalação do Software Readsoft Documents For Invoices	31
4.3.4	Parametrização dos ficheiros de configuração da aplicação	31
4.3.5	Parametrização da aplicação	32
4.3.6	Criação de Nova coluna em Verify	33
4.3.7	Desenvolvimento de uma função que valida se o fornecedor existe para os ficheiros do tipo XML	34
4.3.8	Desenvolvimento em Visual Basic 6 de validações adicionais em Verify (calculo de tolerância)	37
4.3.9	Aplicação de comparação de versões de facturas	41
4.3.10	Desenvolvimento de plugin para transformação de ficheiros Tiff em Multi-tiff e PDF	51
4.3.11	Parametrização do ficheiro eicc.ini para enviar os dados para Cockpit/SAP	54
4.3.12	Parametrização de Cockpit	55

## 4.1 NOTA INTRODUTÓRIA

O caso prático abordado nesta dissertação baseia-se na integração entre a aplicação *Readsoft Documents For Invoices* e a aplicação *Cockpit/SAP* para uma empresa de construção civil que processa anualmente cerca de 315000 facturas. O protótipo que foi desenvolvido tem uma arquitectura um pouco diferente da real devido ao número de máquinas que seriam necessárias.

## 4.2 ARQUITECTURA

Para se entender um pouco melhor a arquitectura de uma plataforma integrada de tratamento electrónico de dados, pode observar-se a figura 17.

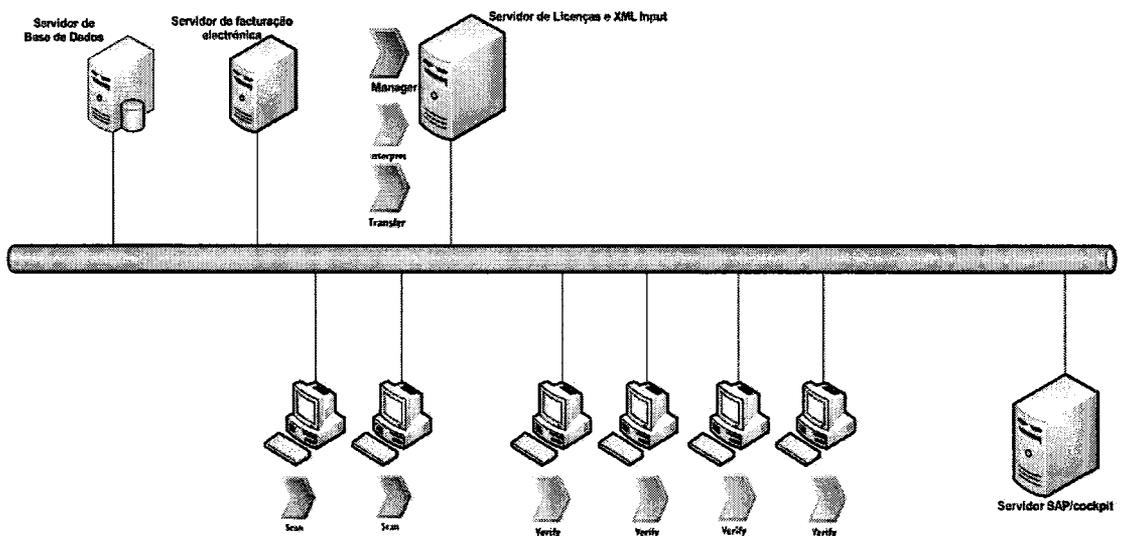


Figura 17 – Arquitectura de aplicação Readsoft integrada com SAP

A arquitectura acima baseia-se em 3 fases:

### 1- Digitalização e interpretação dos documentos:

- 1.1- Os documentos chegam à organização em papel e após serem indexados, são digitalizados no módulo *Scan* pelo perfil pretendido. Durante a digitalização as imagens são guardadas no servidor de licenças. Após a digitalização estar concluída é feito o OCR sobre o documento no módulo *Interpret* e os dados capturados são guardados na base de dados.
- 1.2- Caso os documentos estejam em formato electrónico, o software *XML Input* vai buscar estes documentos ao servidor de facturação electrónica e transforma-os legível pelo *Invoices*. Após este processo

estar terminado, os documentos são interpretados no módulo *Interpret* e os dados capturados são guardados na base de dados.

2- Controlo de qualidade:

2.1- O utilizador efectua o controlo de qualidade dos dados capturados, documento a documento. Este controlo de qualidade é importante para que o utilizador confirme se os dados foram bem capturados pela aplicação e aprove os mesmos para que o documento seja disponibilizado em *Cockpit*.

3- Transferência e Lançamento dos documentos:

3.1- Os dados e a imagem dos documentos são transferidos para *Cockpit*. Em *Cockpit* o utilizador acede documento a documento e lança-os para contabilidade.

## 4.3 INSTALAÇÃO E DESENVOLVIMENTO

A implementação deste projecto baseou-se nas seguintes fases:

### 4.3.1. ANÁLISE E LEVANTAMENTO DE NECESSIDADES

Foram efectuadas diversas abordagens junto do cliente para saber qual a actual realidade do mesmo e a forma de trabalhar, quais as expectativas do cliente em relação ao novo software. Sabendo tudo isto partiu-se para o desenho da arquitectura da solução e desta forma tornou-se mais fácil partir para a implementação do projecto.

### 4.3.2. SEPARAÇÃO DOS DOCUMENTOS E ARQUIVO FÍSICO

Como já foi referido anteriormente, o *Invoices* permite duas formas de separação de documentos com mais que uma página. A separação pode ser feita através de folha em branco ou então através de código de barras.

Neste projecto utilizou-se o código de barras. O código de barras é impresso numa impressora de etiquetas e colado na primeira página de cada factura. Este código para além da separação de documentos, permite uma melhor organização dos documentos a nível de arquivo físico, mas também é

importante para colocar informações adicionais que não vêm na factura, como por exemplo o número da empresa cliente.

### 4.3.3. INSTALAÇÃO DO SOFTWARE READSOFT DOCUMENTS FOR INVOICES 5-5

Após o desenho da aplicação ter sido aprovado partiu-se para a instalação do software nas diferentes máquinas, com as respectivas restrições de acesso aos módulos consoante a máquina onde se está a instalar. Configurou-se o *Microsoft SQL Server 2005* no servidor de base de dados e estabeleceram-se as permissões de acesso através de *SQL Authentication*. Instalou-se também no servidor de licenças o software *Flexlm* que gere a licença do *Readsoft Documents for Invoices*.

### 4.3.4. PARAMETRIZAÇÃO DOS FICHEIROS DE CONFIGURAÇÃO DA APLICAÇÃO

- O ficheiro *Eiglobal.ini* é um ficheiro de parametrização único na arquitectura da aplicação e que se encontra no servidor de licenças. Neste ficheiro definem-se, por exemplo:

- Os parâmetros de acesso à base de dados:

```
[Database]
Type =SQL Server
Version =7.0
Server =IDW-EM\SQLEXPRESS
Databasename =INVOICES
Username =Edgar
Password=017E0033003C00CD00E400F300B6001300FC00B60 [Encriptada]
```

- As características dos scanners que vão ser utilizados pela aplicação;
- Os tipos de moeda que vão ser reconhecidos como válidos pela aplicação;
- Tipos de tabelas utilizadas para captar dados;
- Formatos de imagem suportados;

- O ficheiro *eilocal.ini* é também um ficheiro de parametrização que está presente em todos os computadores pertencentes à arquitectura da aplicação. Neste ficheiro definem-se:
  - Os caminhos da aplicação (imagens, pasta global, binários, etc);
  - Definições do scanner que vai estar ligado a máquina;
  - Chamada dos *plugins* nos diferentes módulos;
  
- O ficheiro *eiglobalextra.ini*, existe apenas no servidor. Os parâmetros definidos neste ficheiro são válidos para toda a aplicação:
  - Taxas de IVA válidas para reconhecimento;
  - Activar ou desactivar validações de campos de IVA, total, valor de base;
  - Formatos suportados de datas.

#### 4.3.5. PARAMETRIZAÇÃO DA APLICAÇÃO

Com a instalação concluída, é necessário criar os perfis e respectivos campos. No início do projecto definem-se que tipos de facturas se vão tratar e que campos interessam captar. Normalmente cada tipo de factura corresponde a um perfil no *Invoices*, perfil esse no qual se vão definir os campos a captar, a forma como esses campos vão ser capturados, o formato dos mesmos, etc.

Depois de concluir a criação do perfil é necessário criar uma “*JobDescription*” que vai estar associado a esse perfil. Nesta “*JobDescription*” definem-se as opções pretendidas para este perfil em cada módulo, tais como:

- Método de separação de facturas;
- “*ScannerDescription*” – Opções de digitalização para este perfil;
- Tipo de origem das facturas a interpretar;
- Definir quais os documentos que se pretende que fiquem disponíveis para verificação;

- Ordem de verificação dos campos;
- Activar ou desactivar interpretação de campos em Verify;
- “TransactionDescription” – Associar um formato de output ao perfil;
- Definir quais os documentos que se pretende que fiquem disponíveis para transferência;
- Localização do ficheiro de output e formato desse mesmo ficheiro (txt, csv, xls...);

#### 4.3.6. CRIAÇÃO DE NOVA COLUNA EM VERIFY

Neste projecto o código de barras que é impresso e colado na primeira página de cada factura tem um papel muito importante, portanto era importante para o utilizador conseguir de alguma forma visualizar esse código de barras antes de abrir o documento em *Verify*, porque é mais fácil identificar uma factura se houver possibilidade de visualizar o código de barras no ecrã principal.

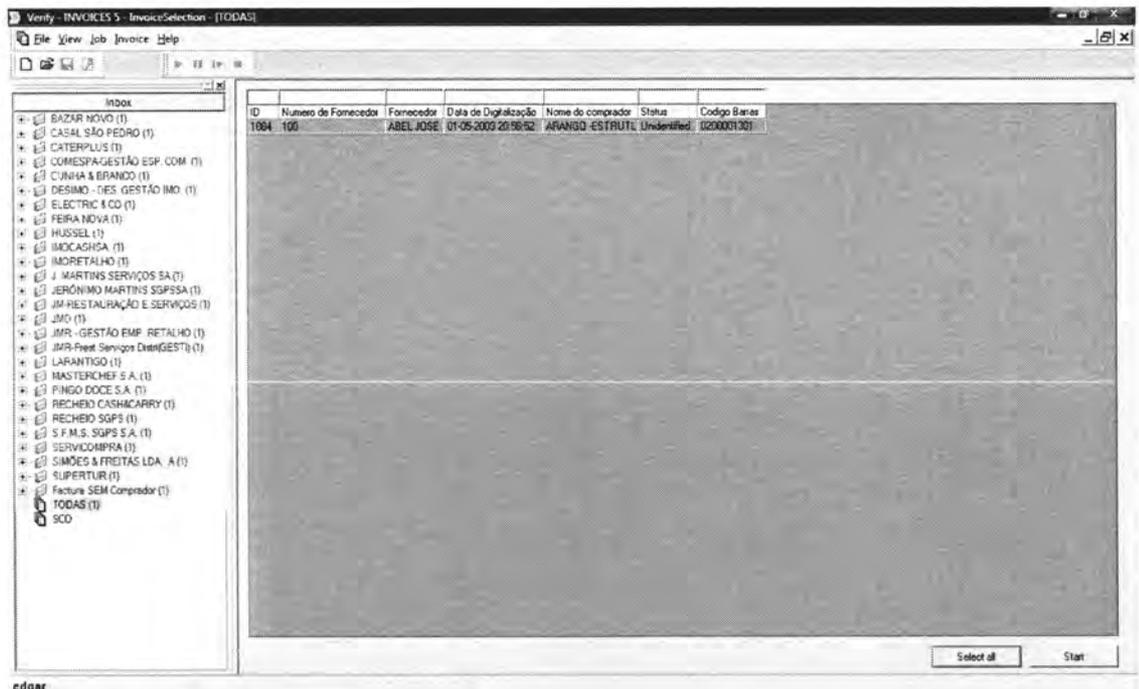


Figura 18 – Coluna adicional em *Verify*.

Para visualizar o código de barras como mostra a figura em cima, é necessário, definir uma variável temporária e em cada interpretação preenche-la:

```
cod_barras = objEHApplication.CurrentInvoice.Fields("Codigo_Barras").Value
```

```
objEHApplication.CurrentInvoice.UserDefinedVariables.Add "Codigo_Barras",  
cod_barras
```

É também necessário mapear essa variável no *Invoices* para que apareça em *Verify*:

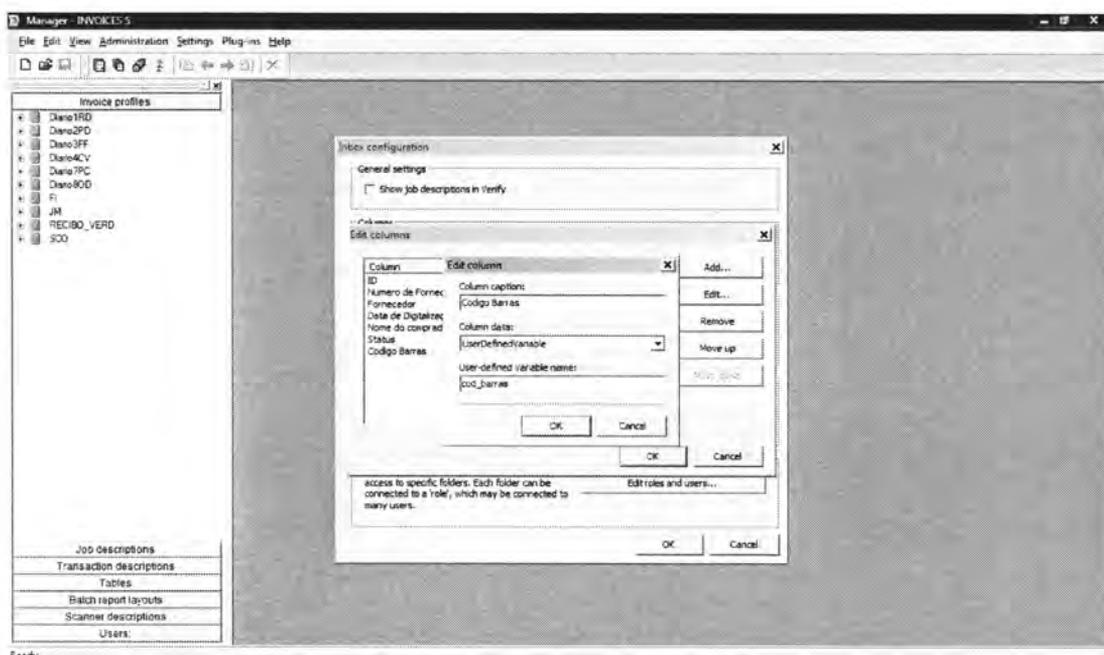


Figura 19 – Mapeamento da coluna no *Manager*.

#### 4.3.7. DESENVOLVIMENTO DE UMA FUNÇÃO QUE VALIDA SE O FORNECEDOR EXISTE PARA OS FICHEIROS DO TIPO XML

Os ficheiros XML que chegam à empresa não trazem na sua estrutura o número de fornecedor da factura que está a ser tratada. Desta forma foi necessário consultar a base de dados do *Invoices* para que através do contribuinte que consta da estrutura do XML obter o número de fornecedor respectivo. Assim desenvolveu-se no módulo *Interpret* o seguinte:

- Primeiro efectuou-se a conexão ao módulo e evento pretendido:

```
Public objEHApplication As EHICOM.Application
```

*Set objEHApplication = objEHApp*

*Dim module*

*module = objEHApplication.ModuleType*

*Select Case module*

*Case eiInterpret*

*objEHApplication.Subscribe Me, "InvoiceInterpreted",  
"OnInvoiceInterpreted"*

- De seguida verificou-se se o documento que está a ser interpretado é do tipo XML e em caso afirmativo procede-se a obtenção do número de fornecedor:

*If (objEHApplication.CurrentInvoice.InvoiceDefinition.InvoiceProfile.Name =  
"XML\_MESP") Then*

*Dim numSupplier As String*

*Dim contribuinte As String*

*contribuinte = objEHApplication.CurrentInvoice.Fields("Contribuinte").Value*

*If (contribuinte <> "") Then*

*numSupplier = GetNumSupplier(contribuinte)*

*Else*

*MsgBox "Fornecedor não existe"*

*End If*

*objEHApplication.CurrentInvoice.Fields("NumeroFornecedor").Value =  
numSupplier*

```

objEHApplication.CurrentInvoice.UserDefinedVariables.Add
"NumFornecedor", numSupplier

End If

Public Function GetNumSupplier(ByVal contribuinte As String) As String

    On Error GoTo GetNumSupplierErro

    Dim sqlQuery

    Dim usr

    Dim usrErr

    Dim lenUsr

    Dim usrCerto

    sqlQuery = "select suppliernumber from zrs_supplier where name2 = " &
contribuinte & ""

    DoConnectDB2

    Set Table.ActiveConnection = ConnectDB

    Set Table = ConnectDB.Execute(sqlQuery)

    usr = Table.GetString

    usrErr = Right$(usr, 1)

    lenUsr = Len(usr)

    usrCerto = Left$(usr, lenUsr - 1)

    DoDisconnectDB2

    GetNumSupplier = usrCerto

Exit Function

```

*GetNumSupplierErro:*

*DoDisconnectDBz*

*MsgBox "Erro na conexao a base de dados"*

*End Function*

Após se obter o número o mesmo vai ser enviado para o campo respectivo do módulo *Verify* e guardado numa variável temporária para posteriormente ser enviado para Cockpit.

#### 4.3.8. DESENVOLVIMENTO EM VISUAL BASIC 6 DE VALIDAÇÕES ADICIONAIS EM VERIFY (CALCULO DE TOLERÂNCIA)

O software *Invoices* possibilita que em *Verify*, após a validação dos dados por parte do utilizador sejam efectuados cálculos para verificar se:

$\text{Valor\_base} + \text{Valor\_IVA} = \text{Total\_Factura}$

Neste projecto foi pedido que se fizesse este cálculo mas para facturas que tivessem mais de uma taxa de IVA, ou para facturas em que fosse necessário subtrair o valor da retenção. Foi por isso necessário desenvolver validações adicionais para efectuarem este cálculo:

- Começou-se novamente por estabelecer a conexão ao módulo e evento pretendido:

*Case eiVerify*

*objEHApplication.Subscribe Me, "InvoiceValidated", "OnInvoiceValidated"*

- De seguida faz-se a leitura dos campos que vão ser utilizados para fazer os cálculos:

*Set TotalObj = objClassicEHApplication.Invoice.getfield("Total\_Fact")*

*Total = TotalObj.GetValueStr()*

*Set Base\_IObj = objClassicEHApplication.Invoice.getfield("Base\_Imp")*

```

Base_I = Base_IObj.GetValueStr()

Set Base_I2Obj = objClassicEHApplication.Invoice.getField("Base_Imp_2")

Base_I2 = Base_I2Obj.GetValueStr()

Set Base_I3Obj = objClassicEHApplication.Invoice.getField("Base_Imp_3")

Base_I3 = Base_I3Obj.GetValueStr()

Set Taxa_IObj = objClassicEHApplication.Invoice.getField("Taxa_Iva")

Taxa_I = Taxa_IObj.GetValueStr()

Set Taxa_I2Obj = objClassicEHApplication.Invoice.getField("Taxa_Iva_2")

Taxa_I2 = Taxa_I2Obj.GetValueStr()

Set Taxa_I3Obj = objClassicEHApplication.Invoice.getField("Taxa_Iva_3")

Taxa_I3 = Taxa_I3Obj.GetValueStr()

Set Valor_IObj = objClassicEHApplication.Invoice.getField("Valor_Iva")

Valor_I = Valor_IObj.GetValueStr()

Set Valor_I2Obj =
objClassicEHApplication.Invoice.getField("Valor_Iva_2")

Valor_I2 = Valor_I2Obj.GetValueStr()

Set Valor_I3Obj =
objClassicEHApplication.Invoice.getField("Valor_Iva_3")

Valor_I3 = Valor_I3Obj.GetValueStr()

Set retencaoObj=
objClassicEHApplication.Invoice.getField("Retencao_Fonte")

retencao = retencaoObj.GetValueStr()

```

- Após ter os dados todos necessários, implementaram-se as validações:

1ª - Verificar se os valores das bases a multiplicar pelas taxas de IVA correspondentes são equivalentes aos valores do IVA. Este cálculo é efectuado com base num limite de tolerância previamente definido:

```
Dim resultado As String
```

```
resultado = (CDec(CCur(bas_aux)) * CDec(taxa_aux)) / 100
```

```
If ((CDec(resultado) <= (CDec(val_aux) + CDec(tolerancia))) And  
(CDec(resultado) >= (CDec(val_aux) - CDec(tolerancia)))) Then
```

```
objEHApplication.CurrentInvoice.Fields("Valor_Iva").Status=  
eiFieldComplete
```

```
Else
```

```
objEHApplication.CurrentInvoice.Fields("Valor_Iva").Status=  
eiFieldValidationError
```

```
MsgBox "O Valor da Base " & CDec(CCur(bas_aux)) & " multiplicado  
pela taxa de iva " & CDec(taxa_aux) & " é diferente do valor do iva " &  
CDec(val_aux)
```

```
End If
```

```
If  
objEHApplication.CurrentInvoiceDefinition.FieldDefinitions("Valor_Iva_2").  
Enabled = True Then
```

```
Dim resultado2 As String
```

```
resultado2 = (CDec(CCur(bas_aux2)) * CDec(taxa_aux2)) / 100
```

```
If ((CDec(resultado2) <= (CDec(val_aux2) + CDec(tolerancia))) And  
(CDec(resultado2) >= (CDec(val_aux2) - CDec(tolerancia)))) Then
```

```
objEHApplication.CurrentInvoice.Fields("Valor_Iva_2").Status =  
eiFieldComplete
```

```
Else
```

```
objEHApplication.CurrentInvoice.Fields("Valor_Iva_2").Status =  
eiFieldValidationError
```

```
MsgBox "O Valor da Base " & CDec(CCur(bas_aux2)) & "  
multiplicado pela taxa de iva " & CDec(taxa_aux2) & " é diferente do valor do iva  
" & CDec(val_aux2)
```

*End If*

*End If*

*If*

*objEHApplication.CurrentInvoiceDefinition.FieldDefinitions("Valor\_Iva\_3").  
Enabled = True Then*

*Dim resultado3 As String*

*resultado3 = (CDec(CCur(bas\_aux3)) \* CDec(taxa\_aux3)) / 100*

*If ((CDec(resultado3) <= (CDec(val\_aux3) + CDec(tolerancia))) And  
(CDec(resultado3) >= (CDec(val\_aux3) - CDec(tolerancia)))) Then*

*objEHApplication.CurrentInvoice.Fields("Valor\_Iva\_3").Status =  
eiFieldComplete*

*Else*

*objEHApplication.CurrentInvoice.Fields("Valor\_Iva\_3").Status =  
eiFieldValidationError*

*MsgBox "O Valor da Base " & CDec(CCur(bas\_aux3)) & "  
multiplicado pela taxa de iva " & CDec(taxa\_aux3) & " é diferente do valor do iva  
" & CDec(val\_aux3)*

*End If*

*End If*

Se o resultado do cálculo for o esperado então o campo do valor do IVA fica no estado “complete”, caso contrário surge uma mensagem no ecrã e o campo fica no estado de erro.

No código acima descrito é utilizada a variável “tolerancia”, esta variável é lida de um ficheiro txt, ou seja, é parametrizável consoante as necessidades do cliente.

2ª - Verificar se a soma de todas as bases com os valores do IVA são iguais ao total da factura:

*If (CDec(tot\_aux) = (CDec(bas\_aux) + CDec(val\_aux) + CDec(bas\_aux2)  
+ CDec(val\_aux2) + CDec(bas\_aux3) + CDec(val\_aux3) - CDec(ret\_aux))) Then*

*tot\_aux = Replace(tot\_aux, "-", "")*

*objEHApplication.CurrentInvoice.Fields("Total\_Fact").Value = tot\_aux*

```
objEHApplication.CurrentInvoice.Fields("Total_Fact").Status =  
eiFieldComplete
```

*Else*

```
objEHApplication.CurrentInvoice.Fields("Total_Fact").Status =  
eiFieldValidationError
```

```
MsgBox "O Valor Total " & tot_aux & " é DIFERENTE das Bases (" &  
CDec(bas_aux) & ") + (" & CDec(bas_aux2) & ") + (" & CDec(bas_aux3) & ") +  
Valores de Iva (" & CDec(val_aux) & ") + (" & CDec(val_aux2) & ") + (" &  
CDec(val_aux3) & ") - retenção na fonte (" & CDec(ret_aux) & ") "
```

*End If*

#### 4.3.9. APLICAÇÃO DE COMPARAÇÃO DE VERSÕES DE FACTURAS

O processo de verificação de facturas requer muita atenção por parte do utilizador, porque não podem haver enganar e a empresa correr o risco de lançar facturas com valores errados. Caso existam enganar é necessário detectar quem cometeu esse erro. Nesse sentido foi desenvolvida uma pequena aplicação "Confirma\_Facturas" (para um perfil específico do Invoices) na qual o administrador pode consultar o histórico dos documentos processados no *Invoices* e verificar que dados foram interpretados no módulo *Interpret* e quais os dados que foram alterados em *Verify* pelo utilizador, bem como o utilizador que efectuou essas alterações. Esta aplicação de foi desenvolvida em C# através da ferramenta Visual Studio 2008 Express Edition.

A aplicação inicia com uma janela de login:

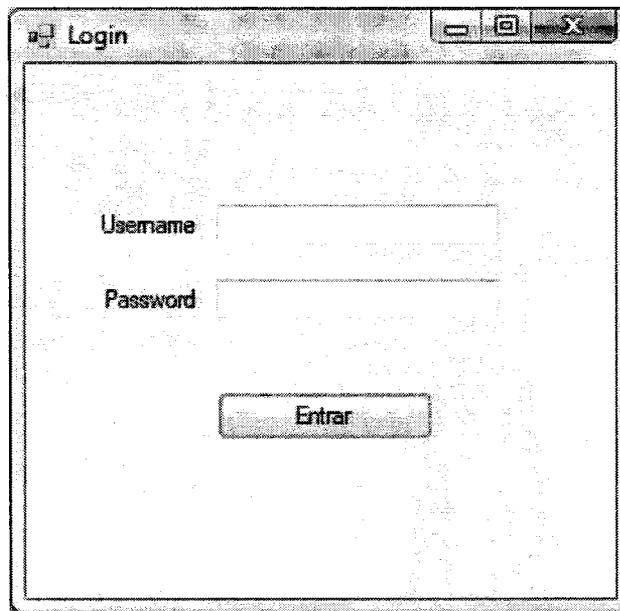


Figura 20 – Janela de login da aplicação “Confirma\_Facturas”.

Nesta janela o utilizador introduz os seus dados de acesso, que vão ser comparados com os que existem na base de dados SQL:

```
private void cmdEntrar_Click(object sender, EventArgs e)
{
    string dbConnString = this.getConnectStr();

    string cmdString = "select * from TAB_LOGIN where Username='" +
this.txtUser.Text + "' AND Password='" + this.txtPass.Text + "'";

    SqlConnection con = new SqlConnection(dbConnString);

    con.Open();

    cmdSelect = con.CreateCommand();

    cmdSelect.CommandText = cmdString;

    SqlDataReader dr = cmdSelect.ExecuteReader();

    if (dr.Read() == false)
    {
        EncDec();

        MessageBox.Show("Dados Incorrectos!");
    }
}
```

```

dr.Close();

    con.Close();

}

Else

{
    dr.Close();

    con.Close();

    EncDec();

    this.Hide();

    string role = verificaRole();

    Confirma_Facturas conf = new Confirma_Facturas();

    conf.Show();

    conf.testaBotao(this.txtUser.Text,role);

}

}

private string verificaRole()

{

    string dbConnString = this.getConnectStr();

    string cmdString = "select Role from TAB_LOGIN where Username=" +
this.txtUser.Text + " AND Password=" + this.txtPass.Text + """;

    SqlConnection con = new SqlConnection(dbConnString);

    con.Open();

    cmdSelect = con.CreateCommand();

    cmdSelect.CommandText = cmdString;

    SqlDataReader dr = cmdSelect.ExecuteReader();

    dr.Read();

```

```

string role = dr.GetSqlString(o).ToString();

dr.Close();

con.Close();

EncDec();

return role;

}

```

Após efectuar a autenticação com sucesso, surge a seguinte janela:

The screenshot shows a window titled 'Confirma\_Facturas' with a 'Facturas' section. It contains two columns of data: 'Interpretadas' and 'Verificadas'. Each column has a 'Código BARRAS' field (0400001302) and a 'Utilizador' field. The 'Verificadas' column has several fields highlighted in red, indicating they have been modified. The 'Utilizador' field in the 'Verificadas' column is highlighted in red and contains the value 'edgar'. Other red fields include 'Base de Imposto 2', 'Taxa 2', 'Valor do Iva 2', 'Base de Imposto 3', 'Taxa 3', 'Valor do Iva 3', 'Status', 'Data do Módulo', 'Hora do Módulo', 'Data Débito Conta', 'Número Prestação', 'Local Consumo', 'Consumo', and 'Retenção Forte'.

Interpretadas		Verificadas	
Código BARRAS	0400001302	Código BARRAS	0400001302
Utilizador	admin	Utilizador	edgar
Base de Imposto 2	0.00	Base de Imposto 2	0.00
Módulo	I	Módulo	T
Taxa 2		Taxa 2	
Número de Fornecedor	3	Número de Fornecedor	30
Valor do Iva 2	0.00	Valor do Iva 2	0.00
Número de Comprador	04	Número de Comprador	04
Base de Imposto 3	0.00	Base de Imposto 3	0.00
Taxa 3		Taxa 3	
Número de Factura		Número de Factura	432522
Valor do Iva 3	0.00	Valor do Iva 3	0.00
Data da Factura		Data da Factura	20091010
Total da Factura	88.35	Total da Factura	88.35
Status	8	Status	11
Número de PO		Número de PO	
Data do Módulo	2010428	Data do Módulo	2010428
Hora do Módulo	20m8m3s	Hora do Módulo	20m14m46s
Número de Contrato		Número de Contrato	
Moeda	EUR	Moeda	EUR
Data Débito Conta		Data Débito Conta	
Número Prestação		Número Prestação	
Tipo de Documento	0	Tipo de Documento	0
Base de Imposto 1	84.951	Base de Imposto 1	84.95
Local Consumo		Local Consumo	
Taxa 1	5.00%	Taxa 1	4.00%
Consumo	Nulo	Consumo	Nulo
Valor do Iva 1	3.40	Valor do Iva 1	3.40
Retenção Forte	0.00	Retenção Forte	0.00

Figura 21 – Janela de verificação de versões da aplicação “Confirma\_Facturas”.

Os valores que estão a vermelho são os valores que foram alterados no módulo de *Verify*. Os campos mais importantes são os campos referentes a montantes dos documentos e o campo utilizador. Na figura acima o utilizador que se logou tem direitos de administração desta aplicação, ou seja, pode criar novo utilizador para aceder a esta aplicação, remover ou alterar password. Um utilizador normal só consegue alterar a própria password e visualizar os dados.

Os dados das facturas que surgem nesta aplicação, são inseridos numa base de dados (que não a do *Invoices*) durante a interpretação e a verificação do

documento. Os dados da coluna esquerda são inseridos durante o módulo *Interpret*, os dados da coluna direita são inseridos no final da verificação do documento em *Verify*. O código em baixo representa a inserção na BD no final da verificação:

*Public Function OnInvoiceApproved() As Long*

*On Error GoTo OnInvoiceApprovedError*

*If (objEHApplication.CurrentInvoiceProfile.Name = "FI") Then*

*DBOpen*

*Dim retencao, nome\_user, Valor\_I, Valor\_I2, Valor\_I3, Nota\_D, Taxa\_I, Taxa\_I2, Taxa\_I3, Base\_I, Base\_I2, Base\_I3, Local\_C, Data\_Deb\_C, N\_prest, Moeda, Tipo\_Doc, N\_PO, N\_Contrato, Data\_F, Cod\_B, Status, Modulo, supp, Total, N\_Factura, buyer, ID\_unico,data,Hora As String*

*Dim SysTime As SYSTEMTIME*

*GetSystemTime SysTime*

*ID\_unico = objEHApplication.CurrentInvoice.Fields("ID").Value*

*Cod\_B = objEHApplication.CurrentInvoice.Fields("Codigo\_Barras"). Value*

*nome\_user = objEHApplication.CurrentUser.Name*

*Modulo = "T"*

*supp = objEHApplication.CurrentInvoiceDefinition.suppliernumber*

*buyer = Left\$(Cod\_B, 2)*

*N\_Factura = objEHApplication.CurrentInvoice.Fields("Fact\_Num"). Value*

*N\_Factura = Replace(N\_Factura, "", "")*

*Data\_F = objEHApplication.CurrentInvoice.Fields("Data\_Fact"). Value*

*Data\_F = Replace(Data\_F, "", "")*

*Total = objEHApplication.CurrentInvoice.Fields("Total\_Fact").Value*

*Total = Replace(Total, "", "")*

*N\_PO = objEHApplication.CurrentInvoice.Fields("Encom\_Num"). Value*

*N\_PO = Replace(N\_PO, "", "")*

*N\_Contrato=objEHApplication.CurrentInvoice.Fields("Contrato\_Nu  
m").Value*

*N\_Contrato = Replace(N\_Contrato, "", "")*

*Moeda = objEHApplication.CurrentInvoice.Fields("Moeda").Value*

*Moeda = Replace(Moeda, "", "")*

*Nota\_D = objEHApplication.CurrentInvoice.Fields("Nota\_Debito"). Value*

*Nota\_D = Replace(Nota\_D, "", "")*

*Base\_I = objEHApplication.CurrentInvoice.Fields("Base\_Imp"). Value*

*Base\_I = Replace(Base\_I, "", "")*

*Base\_I2 = objEHApplication.CurrentInvoice.Fields("Base\_Imp\_2"). Value*

*Base\_I2 = Replace(Base\_I2, "", "")*

*Base\_I3 = objEHApplication.CurrentInvoice.Fields("Base\_Imp\_3"). Value*

*Base\_I3 = Replace(Base\_I3, "", "")*

*Taxa\_I = objEHApplication.CurrentInvoice.Fields("Taxa\_Iva"). Value*

*Taxa\_I = Replace(Taxa\_I, "", "")*

*Taxa\_I2 = objEHApplication.CurrentInvoice.Fields("Taxa\_Iva\_2"). Value*

*Taxa\_I2 = Replace(Taxa\_I2, "", "")*

*Taxa\_I3 = objEHApplication.CurrentInvoice.Fields("Taxa\_Iva\_3"). Value*

*Taxa\_I3 = Replace(Taxa\_I3, "", "")*

*Valor\_I = objEHApplication.CurrentInvoice.Fields("Valor\_Iva"). Value*

*Valor\_I = Replace(Valor\_I, "", "")*

*Valor\_I2 = objEHApplication.CurrentInvoice.Fields("Valor\_Iva\_2"). Value*

*Valor\_I2 = Replace(Valor\_I2, "", "")*

*Valor\_I3 = objEHApplication.CurrentInvoice.Fields("Valor\_Iva\_3"). Value*

Valor\_I3 = Replace(Valor\_I3, "", "")

Status = objEHApplication.CurrentInvoice.Status

data = SysTime.wYear & SysTime.wMonth & SysTime.wDay

Hora = SysTime.wHour & "m" & SysTime.wMinute & "m" & SysTime.wSecond & "s"

Data\_Deb\_C = objEHApplication.CurrentInvoice.Fields("Data\_Deb\_Conta").Value

Data\_Deb\_C = Replace(Data\_Deb\_C, "", "")

N\_prest = objEHApplication.CurrentInvoice.Fields("N\_Prestacao").Value

N\_prest = Replace(N\_prest, "", "")

Local\_C = objEHApplication.CurrentInvoice.Fields("Local\_Consumo").Value

Local\_C = Replace(Local\_C, "", "")

retencao = objEHApplication.CurrentInvoice.Fields("Retencao\_Fonte").Value

retencao = Replace(retencao, "", "")

ConADO.Execute ("INSERT INTO TAB\_LOG\_FI\_V (id\_tab,Codigo\_Barras, Modulo, Num\_Fornecedor, NumComprador, Status, Data\_modulo, Hora\_modulo, Num\_Factura, Data\_Factura, Num\_PO, Num\_Contrato, Moeda, Tipo\_Doc, Base\_imp\_1, Taxa\_1, Valor\_iva\_1, Base\_imp\_2, Taxa\_2, Valor\_iva\_2, Base\_imp\_3, Taxa\_3, Valor\_iva\_3, Data\_Deb\_Conta, Numero\_Prestacao, Local\_Consumo, Utilizador, Total\_factura, Retencao\_Fonte) VALUES ('" & ID\_unico & "','" & Cod\_B & "','" & Modulo & "','" & supp & "','" & buyer & "','" & Status & "','" & data & "','" & Hora & "','" & N\_Factura & "','" & Data\_F & "','" & N\_PO & "','" & N\_Contrato & "','" & Moeda & "','" & Nota\_D & "','" & Base\_I & "','" & Taxa\_I & "','" & Valor\_I & "','" & Base\_I2 & "','" & Taxa\_I2 & "','" & Valor\_I2 & "','" & Base\_I3 & "','" & Taxa\_I3 & "','" & Valor\_I3 & "','" & Data\_Deb\_C & "','" & N\_prest & "','" & Local\_C & "','" & nome\_user & "','" & Total & "','" & retencao & "'"))

DBCclose

End If

OnInvoiceApproved = evtOK

Exit Function

OnInvoiceApprovedError:

MsgBox "Erro no evento" & vbNewLine & Err.Description

End Function

Após a inserção ser efectuada os dados ficam disponíveis para consulta na aplicação “Confirma\_Facturas”. A aplicação “Confirma\_Facturas” utiliza três tabelas:

- TAB\_LOG\_FI\_I – Nesta tabela são guardados os dados de *Interpret*;
- TAB\_LOG\_FI\_V – Nesta tabela são guardados os dados de *Verify*;
- TAB\_LOGIN – Esta tabela contém a listagem de utilizadores que acedem à aplicação;

Existe ainda uma *View* que é utilizada para a aplicação “Confirma\_Facturas” consultar a informação das duas tabelas.

Como esta aplicação iria ficar em computadores de administradores ou pessoas externas à aplicação, houve necessidade de esconder os dados de acesso à base de dados. Os dados de acesso à base de dados estão guardados num ficheiro XML com a seguinte estrutura:

```
<note>
  <dbConnection>Initial Catalog=LOG_JMARTINS;Data Source=SERVER;User
  ID=USER;Password=PASS;</dbConnection>
</note>
```

Foi necessário encriptar estes dados do ficheiro e cada vez que se faz o login esses dados são descriptados para serem lidos e voltam a ser encriptados. As funções de encriptação de descriptação utilizadas foram as seguintes:

```
public static void Encrypt(XmlDocument Doc, string ElementName,
  SymmetricAlgorithm Key)
  {
    if (Doc == null)
      throw new ArgumentNullException("Doc");

    if (ElementName == null)
      throw new ArgumentNullException("ElementToEncrypt");

    if (Key == null)
      throw new ArgumentNullException("Alg");
```

```

        XmlElement elementToEncrypt =
Doc.GetElementsByTagName(ElementName)[0] as XmlElement;

        if (elementToEncrypt == null)

        {

            throw new XmlException("The specified element was not found");

        }

        EncryptedXml eXml = new EncryptedXml();

        byte[] encryptedElement = eXml.EncryptData(elementToEncrypt, Key,
false);

        EncryptedData edElement = new EncryptedData();

        edElement.Type = EncryptedXml.XmlEncElementUrl;

        string encryptionMethod = null;

        if (Key is TripleDES)

        {

            encryptionMethod = EncryptedXml.XmlEncTripleDESUrl;

        }

        else if (Key is DES)

        {

            encryptionMethod = EncryptedXml.XmlEncDESUrl;

        }

        if (Key is Rijndael)

        {

            switch (Key.KeySize)

            {

                case 128:

                    encryptionMethod = EncryptedXml.XmlEncAES128Url;

```

```
        break;

    case 192:

        encryptionMethod = EncryptedXml.XmlEncAES192Url;

        break;

    case 256:

        encryptionMethod = EncryptedXml.XmlEncAES256Url;

        break;

    }

}

Else

{

    throw new CryptographicException("The specified algorithm is not
supported for XML Encryption.");

}

    edElement.EncryptionMethod = new
EncryptionMethod(encryptionMethod);

    edElement.CipherData.CipherValue = encryptedElement;

    EncryptedXml.ReplaceElement(elementToEncrypt, edElement, true);

    Doc.Save("Confirma_Facturas.xml");

}

public static void Decrypt(XmlDocument Doc, SymmetricAlgorithm Alg)

{

    if (Doc == null)

        throw new ArgumentNullException("Doc");

    if (Alg == null)
```

```

        throw new ArgumentNullException("Alg");

        XmlElement encryptedElement =
Doc.GetElementsByTagName("EncryptedData")[0] as XmlElement;

        if (encryptedElement == null)

        {

            throw new XmlException("The EncryptedData element was not
found.");

        }

        EncryptedData edElement = new EncryptedData();

        edElement.LoadXml(encryptedElement);

        EncryptedXml exml = new EncryptedXml();

        byte[] rgbOutput = exml.DecryptData(edElement, Alg);

        exml.ReplaceData(encryptedElement, rgbOutput);

        Doc.Save("Confirma_Facturas.xml");

    }

```

Desta forma garante-se que quem utiliza esta aplicação não tem acesso aos parâmetros de acesso à base de dados.

#### 4.3.10. DESENVOLVIMENTO DE PLUGIN PARA TRANSFORMAÇÃO DE FICHEIROS TIFF EM MULTI-TIFF E PDF

A aplicação Readsoft Documents For Invoices gera ficheiros de imagem em formato tif, que após a transferência podem ser transformados para o formato jpg, bmp ou png. Cada página do documento gera um ficheiro tif, no entanto neste projecto foi necessário converter os ficheiros tiff para multi-tiff para pdf e só depois enviá-los para SAP;

Para efectuar este desenvolvimento foi necessário primeiro que tudo validar em que módulo fazia sentido desenvolver esta funcionalidade. Optou-se por desenvolver esta funcionalidade no módulo *Transfer*, porque é neste módulo

que os documentos estão tratados e são transferidos para Cockpit. Assim iniciou-se por estabelecer a conexão ao módulo e evento correcto:

*Case eiTransfer*

```
objEHApplication.Subscribe Me, "InvoiceTransfer", "OnInvoiceTransfer"
```

De seguida foi necessário implementar dentro do evento que se está a utilizar uma forma de converter o tiff para pdf. A passagem de tif para pdf é efectuada em duas fases. Na primeira fase converte-se o tiff para multi-tiff e só depois se converte o multi-tiff para PDF.

Para transformar o ficheiro de imagem em multi-tiff implementou-se o seguinte:

- Utilizou-se o método *ExportToMultiTiff* disponível na API da Readsoft para converter o ficheiro tiff em multi-tiff. Para ser possível efectuar esta conversão é necessário atribuir o nome ao ficheiro multi-tiff que vai ser gerado. O nome que vai ser atribuído, é o mesmo que nome do ficheiro da primeira página da factura seguido de "M".

```
Private Sub ExportToMultiTiff()
```

```
Dim obInvoice As EHICOM.Invoice
```

```
Set obInvoice = objEHApplication.CurrentInvoice
```

```
Dim sTiffname As String
```

```
sTiffname = GetMtiffFileName
```

```
obInvoice.ExportToMultiTiff sTiffname  
ExportToPDF (sTiffname)
```

```
End Sub
```

```
Private Function GetMtiffFileName() As String
```

```
On Error GoTo GetMtiffFileNameError
```

```
Dim mTiffname As String
```

```
Dim imagefile As String
```

```
imagefile = objClassicEHApplication.Invoice.GetImageFile()
```

```
mTiffname = Replace(imagefile, ".TIF", "M.TIF")
```

```
GetMtiffFileName = mTiffname
```

```
GetMtiffFileNameError:
```

```
MsgBox "Erro no evento" & vbNewLine & Err.Description
```

```
End Function
```

Após a conversão para multi-tiff estar concluída, é efectuada a conversão para pdf através da função *ExportToPDF*. Este método é um método que faz parte do ficheiro dll "BT2P.dll" que já existia na IDW. Desta forma utilizou-se esta dll para efectuar a conversão, para tal declarou-se a função que era necessária:

```
Private Declare Function t2p_mx Lib "BT2P.dll" _  
    (ByVal multiple As String, ByVal Out As String) As Long
```

E posteriormente executou-se dentro da função *ExportToPDF*:

```
Private Function ExportToPDF(ByVal sTiffname As String)
```

```
    On Error GoTo ExportToPDFError
```

```
    Dim arg1 As String
```

```
    Dim arg2 As String
```

```
    Dim arg3 As String
```

```
        arg1 = sTiffname + Chr(0) + Chr(0)
```

```
        arg2 = sTiffname
```

```
        arg3 = t2p_mx(arg1, arg2)
```

```
sTiffname = Replace(sTiffname, ".tif", ".pdf")
```

```
    sTiffname = Replace(sTiffname, ".tiff", ".pdf")
```

```
sTiffname = Replace(sTiffname, ".TIF", ".pdf")
```

```
sTiffname = Replace(sTiffname, ".TIFF", ".pdf")
```

```
objEHApplication.CurrentInvoice.UserDefinedVariables.Add  
"EICCARCIMAGEFILE", sTiffname
```

```
ExportToPDFError:
```

```
MsgBox "Erro no evento" & vbCrLf & Err.Description
```

```
End Function
```

A função *t2p\_mx* recebe dois argumentos do tipo string. O primeiro (arg1) consiste no ficheiro que se vai converter em pdf seguido de Chr(o) para terminar a *string*. O segundo argumento (arg2) é o nome do ficheiro pdf. Após ser efectuada a conversão o nome do ficheiro pdf e o respectivo caminho são guardados numa variável temporária do *Invoices* "EICCARCIMAGEFILE" para que quando for efectuada a comunicação com cockpit para a transferência do ficheiro de imagem, seja o ficheiro pdf a ser transferido.

#### 4.3.11. PARAMETRIZAÇÃO DO FICHEIRO EICC.INI PARA ENVIAR OS DADOS PARA COCKPIT/SAP;

O módulo *Transfer* quando efectua a transferência das facturas, liga-se directamente ao *Cockpit* para onde envia os dados tratados e respectiva imagem. Para que esta transferência seja bem sucedida é necessário previamente efectuar várias parametrizações no ficheiro *eicc.ini*:

- É necessário indicar a aplicação para onde se pretendem enviar os dados que foram colocados em variáveis temporárias, ou campos que se pretendam enviar para um campo de *Cockpit* que não o standard:

```
_comp_code=^CODIGO_BARRAS
```

```
_vendor_no=%#SupplNumber
```

```
_currency=%#Currency
```

```
_invoice=&Factura/Credito
```

```
_Content=&Contentor
```

```
_Via_Par=&Viagem_Partida
```

```
_import_export=&Imp_Exp
```

`_tipo_doc=&TIPODOCUMENTO`  
`_GL_CONTA=^GL_conta`  
`_Codigo_Especial=^Codigo_Especial`  
`_Chave_Lancamento=^Chave_Lancamento`  
`_username=&USER`

Os campos do lado esquerdo do igual são os campos que vão ser enviados para *Cockpit*, os campos à direita, são os campos provenientes do *Invoices*;

- É necessário indicar à aplicação os dados de acesso ao servidor de *SAP* (*Cockpit*);
- Indicar qual o formato de imagem que vai ser enviado para *SAP*;
- Indicar qual a versão do sistema *SAP*;

#### 4.3.12. PARAMETRIZAÇÃO DE COCKPIT

Para que os documentos e os respectivos dados cheguem correctamente a *Cockpit* é necessário fazer também algumas parametrizações no *Cockpit*. Estas parametrizações passam por mapeamento de campos ou até mesmo criação de novos campos. A criação de novos campos só é necessária quando estes não existem na versão standard. Um exemplo de um campo que não existe na versão standard é o código de barras, nestas situações é necessário criar campos através da utilização de um tabulador específico de *Cockpit*. Estes campos normalmente servem apenas para consulta em *Cockpit*, os quais, após os documentos serem lançados deixam de ser necessários.

Nos restantes casos em que apenas é necessário mapear campos já existentes, acede-se à transacção */Cockpit/c1* e lá definem-se os mapeamentos:

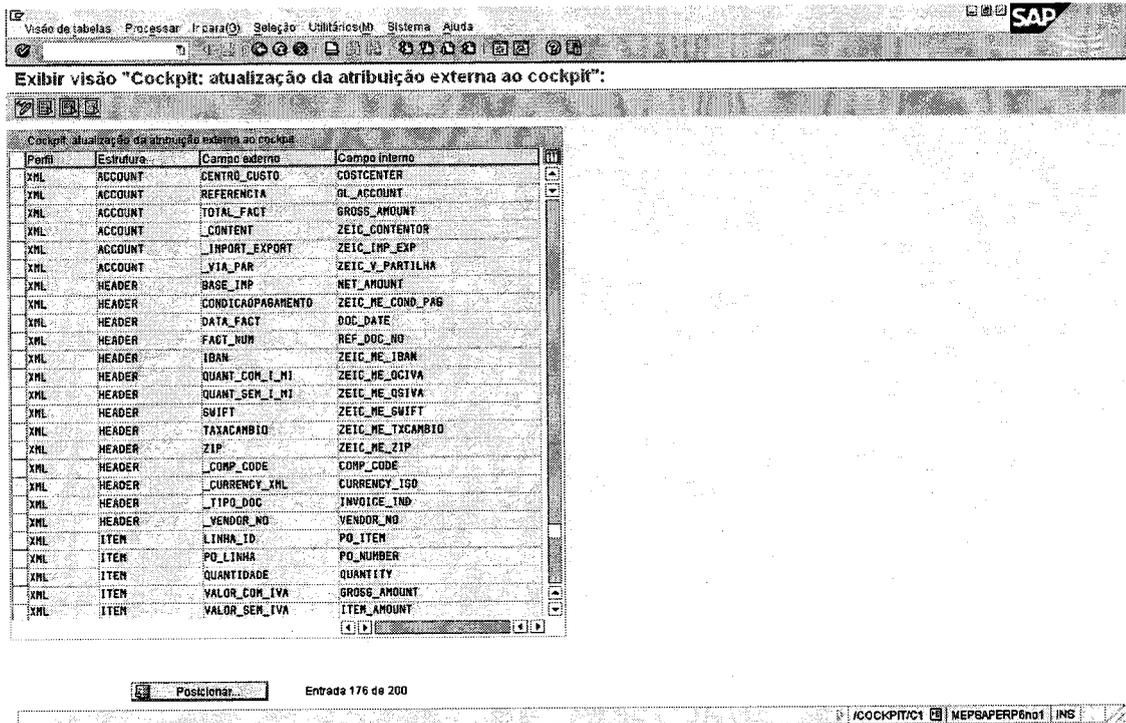


Figura 22 – Mapeamento de campos em Cockpit.

No mapeamento existem quatro colunas para preencher:

- Perfil – Nome do perfil a que pertence o campo que vamos mapear. Este perfil tem que ter o mesmo nome do perfil do *Invoices*.
- Estrutura – Tipo do campo em Cockpit, por exemplo, caso seja um campo de cabeçalho é do tipo *Header*, se for um campo de detalhe é do tipo *Item*.
- Campo externo – Designação do campo que vem do *Invoices*, ou seja, são os nomes previamente definidos no *Invoices* e que estão parametrizados na *TransactionDescription* ou no *Eicc.ini*.
- Campo interno – Designação do campo em *Cockpit*.

Desta forma a aplicação sabe exactamente onde escrever os campos que vêm do *Invoices*. Na figura seguinte pode ver-se o exemplo de um documento proveniente do *Invoices*:

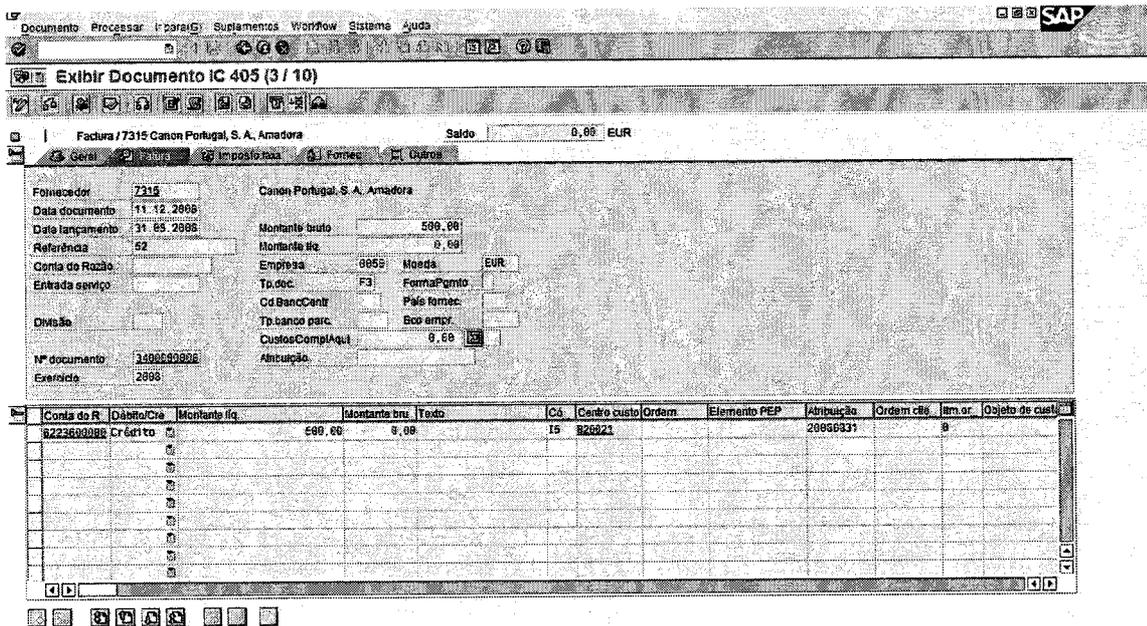


Figura 23 – Detalhe de uma factura em Cockpit.

Para além do mapeamento dos campos é também possível efectuar desenvolvimentos específicos utilizando a linguagem de programação ABAP. Estes desenvolvimentos são efectuados dentro de *User Exits* (eventos de Cockpit). No caso deste projecto foi efectuado um pequeno desenvolvimento que faz a leitura do número de fornecedor da factura que está a ser transferida para Cockpit e através desse número é efectuado um *Select* à tabela “ZCOCKDCL” para se obter o código da conta razão associado a esse fornecedor:

*if not c\_str\_header-ZEIC\_ME\_CAPITAL is initial.*

*DATA: CONTA1AUX LIKE ZCOCKDCL-CONTA1.*

*DATA: CONTA2AUX LIKE ZCOCKDCL-CONTA2.*

*select SINGLE CONTA1 FROM ZCOCKDCL INTO CONTA1AUX where LIFNR =*  
*C\_str\_header-vendor\_no.*

*select SINGLE CONTA2 FROM ZCOCKDCL INTO CONTA2AUX where LIFNR =*  
*C\_str\_header-vendor\_no.*

*MOVE c\_str\_header-ZEIC\_ME\_CAPITAL TO c\_tab\_account-net\_amount.*

*MOVE CONTA1AUX TO c\_tab\_account-gl\_account.*

*modify c\_tab\_account index 1.*

*MOVE c\_str\_header-ZEIC\_ME\_JUROS TO c\_tab\_account-net\_amount.*

*MOVE CONTA2AUX TO c\_tab\_account-gl\_account.*

*APPEND c\_tab\_account.*

*MOVE c\_str\_header-ZEIC\_ME\_DESPBANC TO c\_tab\_account-net\_amount.*

*MOVE '6881006808' TO c\_tab\_account-gl\_account.*

*APPEND c\_tab\_account.*

*endif.*

*if not c\_str\_header-ZEIC\_ME\_APOLICE IS INITIAL.*

*DATA: periodo like c\_tab\_account-text.*

*CONCATENATE 'Apólice - ' c\_str\_header-ZEIC\_ME\_APOLICE ' Período: '  
c\_str\_header-ZEIC\_ME\_PERIODO INTO periodo.*

*MOVE periodo to c\_tab\_account-text.*

*modify c\_tab\_account index 1.*

*endif.*

# 5

## Conclusões

---

5	Conclusões	59
5.1	Apreciação Crítica / Avaliação	60
5.2	Conclusões Finais	61

## 5.1 APRECIÇÃO CRÍTICA / AVALIAÇÃO

A experiência de integração numa equipa jovem de uma empresa pequena permite que seja efectuada esta apreciação crítica a todo o processo a sistemas de capturas de dados:

- Visto tratar-se de uma empresa pequena, o acompanhamento foi efectuado de forma mais atenta, o que possibilita uma melhor aprendizagem. No entanto foi muito complexo perceber a forma de trabalhar e a real vantagem da utilização deste software, esta dificuldade prendeu-se com o facto do total desconhecimento acerca dos processos que envolvem o lançamento de uma simples factura. Após a fase de aprendizagem, proporcionou-se a implementação deste projecto extremamente interessante.
- Na fase de planeamento/enquadramento, foram efectuadas várias reuniões em conjunto com o cliente para que a solução que se ia implementar cumprisse as expectativas do cliente. Nestas reuniões foi notório o desconforto de alguns colaboradores em relação à solução de captura de dados. Este desconforto estava relacionado com o facto de que até a data estas pessoas introduziam os dados manualmente no ERP, sendo este o trabalho diário delas. Com a implementação deste projecto existia o receio por parte destes colaboradores de que passariam a ser desnecessários à empresa o que provocaria o seu despedimento. Foi uma árdua tarefa conseguir tranquilizar estes colaboradores e mostrar-lhes que a aplicação não iria prejudicá-los, mas sim ajudá-los a desempenhar o seu trabalho. Este é um aspecto muito delicado e é também dos mais importantes, porque se existir uma boa relação software/utilizador o sucesso de implementação será maior.
- Durante a implementação do projecto, foram encontrados alguns obstáculos, obstáculos esses que estavam principalmente relacionados, com a forma de trabalhar do cliente e à qual a aplicação se teve de ajustar. O software Readsoft Documents For Invoices tem grande vantagem em relação a outros sistemas do mercado neste aspecto, dado que possibilita desenvolver código adicional em qualquer módulo e porque permite uma integração transparente para o utilizador em SAP.

- O Cockpit é também uma ferramenta poderosa, porque é uma aplicação com um frontend equivalente ao de SAP e nesse sentido não é necessária grande adaptação do utilizador à aplicação.
- O projecto encontra-se actualmente em produção e a processar cerca de 26000 documentos por mês e com perspectivas de aumento de processamento nos próximos anos.

## **5.2 CONCLUSÕES FINAIS**

Na época em que vivemos, a facturação é cada vez mais feita ao segundo, não podem existir atrasos ou extravio de documentos. Nesse sentido cada vez mais empresas optam por este tipo de soluções, no entanto não é só por estes motivos que as empresas estão a aderir a este tipo de tecnologias. Os próprios auditores externos que por vezes visitam as empresas, levam em conta o facto de a empresa utilizar ou não estas aplicações.

A tendência é que cada vez existam mais soluções disponíveis no mercado e as empresas as utilizem. É também notório a evolução da facturação electrónica, são cada vez mais os fornecedores a enviar os documentos em formato electrónico. Existem no entanto ainda muitos fornecedores que enviam facturas manuscritas, ou impressas com muito pouca qualidade, esta é uma situação cada vez menos usual, mas que ainda surge com muita frequência.

Uma solução de captura de dados integrada com o ERP SAP é de facto uma solução muito interessante para as empresas, é no entanto muito dispendiosa e que não está ao alcance de todas. Existem no entanto outras possibilidades de integração com outros ERP's mais baratos mas que como é óbvio não são tão completos.



# 6

## Bibliografia

---

LIMA E.(2002). *C# E .NET - GUIA DO DESENVOLVEDOR*. Rio de Janeiro: Campos.

Williams G. C.(2008). *Implementing SAP ERP Sales & Distribution*. Boston: McGraw-Hill.

IDW - Integration, Development and Warehousing. Obtido em <http://www.idw.pt>

Readsoft International. Obtido em <http://www.readsoft.com/>

SAP. Obtido em <http://www.sap.com/portugal/index.epx>

SAP ERP. Obtido em [http://pt.wikipedia.org/wiki/SAP\\_ERP](http://pt.wikipedia.org/wiki/SAP_ERP)

Como encriptar e desencriptar um ficheiro utilizando o Visual C#. Obtido em <http://support.microsoft.com/kb/307010>

File Encryption/Decryption with Hash Verification in C#  
<http://www.codeproject.com/KB/security/fileencryptdecrypt.aspx>