

António Ricardo Guirruço

**Desenvolvimento e Implementação de Sistema de Facturação:**

**Caso da empresa COSYS, Lda**

Licenciatura em Informática

Universidade Pedagógica de Maputo  
Maputo  
2022

António Ricardo Guirruço

**Desenvolvimento e Implementação de Sistema de Facturação:**

**Caso da empresa COSYS, Lda**

Monografia apresentada ao curso de Informática,  
Faculdade de Engenharias e Tecnologias da  
UPM, para a obtenção do grau académico de  
Licenciatura em Informática com Habilitações  
em Engenharia de Desenvolvimento de Sistemas.

Supervisora:

Mestre Cláudia Ivete F. Jovo

Universidade Pedagógica de Maputo

Maputo

2022

## Índice

LISTA DE TABELAS .....	IV
LISTA DE FIGURAS.....	V
SIGLAS E ABREVIATURAS .....	VI
Declaração.....	IX
Dedicatória.....	X
Agradecimentos .....	XI
RESUMO.....	XII
ABSTRACT.....	XIII
CAPÍTULO I - INTRODUÇÃO.....	1
1.1. Justificativa .....	2
1.2. Problematização.....	2
1.3. Objectivos .....	3
1.3.1. Objectivo Geral.....	3
1.3.2. Objectivos Específicos.....	4
1.4. Questões de Pesquisa .....	4
1.5. Hipóteses.....	4
1.6. Importância do Tema .....	5
1.7. Metodologia do Projecto.....	6
1.7.1. Técnicas de Recolha de Dados .....	6
1.7.2. Metodologia de Desenvolvimento da Aplicação .....	7
1.7.3. Campo de Estudo .....	7

CAPÍTULO II – ENQUADRAMENTO TEÓRICO .....	9
2.1. Facturação .....	9
2.1.1. Tipos de Factura.....	9
2.1.2. Elementos de uma Factura .....	10
2.2. Tecnologias de Informação e Comunicação .....	11
2.3. Arquitectura dos Sistemas Digitas de Informação e Comunicação .....	13
2.3.1. Base de Dados .....	13
2.3.1.1. Função da base de dados .....	15
2.3.1.2. Modelos de base de dados.....	15
2.3.2. WebServer.....	16
2.3.2.1. Princípios de Funcionamento de um <i>WebServer</i> .....	16
2.4. Desenvolvimento de Sistemas Digitas de Informação e Comunicação.....	20
2.4.1. Ciclo de vida de Um Sistema Informático .....	20
2.4.1.1. Modelo Iterativo e Incremental.....	21
2.4.1.2. Fases do Modelo Incremental .....	21
2.4.1.3. Vantagens do Modelo Incremental .....	22
2.4.1.4. Desvantagens do Modelo Incremental.....	22
2.4.2. Ferramentas de Desenvolvimento de Softwares .....	22
2.4.2.1. UML.....	22
2.4.2.2. NetBeans .....	24
CAPÍTULO III – APRESENTAÇÃO DE RESULTADOS.....	26
3.1. Apresentação do Sistema .....	26

3.1.1. Requisitos do Sistema.....	26
3.2. Modelagem do Sistema.....	28
3.2.1. Diagrama dos Principais Casos de Uso .....	28
3.2.1.1 Descrição dos Principais Casos de Uso .....	29
3.2.2. Diagrama de Classes.....	30
3.3. Funcionamento do Sistema.....	31
3.3.1. Modelo de Codificação e Estruturação.....	31
3.3.2. Principais Formulários.....	31
3.3.2.1. Navegação e Acções .....	34
3.4. Segurança do Sistema .....	35
4. CONCLUSÃO.....	37
5. RECOMENDAÇÕES.....	39
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	40
7. APÊNDICES.....	43
8. ANEXOS .....	61

**LISTA DE TABELAS**

<i>Tabela 1 : Requisitos Não-Funcionais Software .....</i>	<i>27</i>
<i>Tabela 2 : Requisitos Não-Funcionais Software 2 .....</i>	<i>28</i>
<i>Tabela 3 : Descrição dos Casos de Uso .....</i>	<i>29</i>

**LISTA DE FIGURAS**

Figura 1 : Modelo Conceitual do Webservice .....	17
Figura 2 : Modelo Incremental .....	21
Figura 3 : Diagrama de Casos de Uso.....	29
Figura 4 : Diagrama de Classes .....	30
Figura 5 : <i>Login</i> .....	31
Figura 6 : Painel Principal.....	32
Figura 7 : Formulário de Registo de Usuário.....	33
Figura 8 : Formulário de Emissão de Factura.....	34
Figura 9 : Campo de Pesquisa.....	35
Figura 10 : Camada de Botões .....	35

## **SIGLAS E ABREVIATURAS**

API - Application Programming Interface.

CASE - Computer-Aided Software Engineering.

CIVA - Código do Imposto sobre o Valor Acrescentado.

CPU - Central Processing Unit.

CRUD - Create, Read, Update and Delete.

CSS3 - Cascading Style Sheets 3<sup>rd</sup> version.

CVS - Concurrent Versions System.

DDR3 - Double Data Rate 3.

EJB - Enterprise JavaBeans.

ERP - Enterprise Resource Planning.

GB - Gigabyte.

GHz - Gigahertz.

HTML - Hypertext Markup Language.

HTTP - Hypertext Transfer Protocol.

IDE – Integrated Development Environment.

IMAP - Internet Message Access Protocol.

IVA - Imposto sobre o Valor Acrescentado.

kB - Kilobyte.

JSP - JavaServer Pages.

JSTL - JavaServer Pages Standard Tag Library.

MD5 - Message-Digest algorithm 5.

MVC – Model –View-Controller.

NASA - National Aeronautics and Space Administration.

NATO - North Atlantic Treaty Organization.

NNTP - Network News Transfer Protocol.

NUIT - Número Único de Identificação Tributária.

PDO – PHP Data Objects.

PHP – Hipertext Processor.

PLPGSQL - Procedural Language/PostgreSQL.

PL/PgSQL - Procedural Language/PostgreSQL.

POP3 - Post Office Protocol 3<sup>rd</sup> version.

RAM - Random Access Memory.

RUP – Rational Unified Process.

SGBD – Sistema de Gestão de Base de Dados.

SNMP - Simple Network Management Protocol.

SQL – Structured Query Language.

SSD - Solid State Drive.

SSL - Secure Sockets Layer.

TIC – Tecnologia de Informação e Comunicação.

UML - Unified Modeling Language.

URL - Uniform Resource Locator.

### **Declaração**

Declaro que esta Monografia é resultado da minha investigação pessoal e das orientações da minha supervisora, o seu conteúdo é original e todas as fontes consultadas estão devidamente mencionadas no texto, nas notas e na bibliografia final.

Declaro ainda que este trabalho não foi apresentado em nenhuma outra instituição para obtenção de qualquer grau académico.

Maputo, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

\_\_\_\_\_

António Ricardo Guirruço

### **Dedicatória**

Dedico este trabalho à minha esposa e filho, Vânia da Mira Afonso e Tarim Osnof Guirruço, as pessoas mais importantes da minha vida, que pela sua existência e sorriso contagiante, mantêm-me firme e dão-me forças para continuar a ser e almejar o melhor que há na vida. Minhas fontes de inspiração e foco.

## **Agradecimentos**

À Deus, pois dele temos o prazer de estar neste mundo e desfrutar das belas coisas que nos são constituídas, além de que Ele nos protege e nos guarda.

À minha esposa e companheira de uma vida inteira, Vânia da Mira Afonso, pelo carinho, amor, amizade e pelos vários momentos, bons e maus, em que sempre estive do meu lado (e espero que sempre esteja).

Ao meu filho e amigo, Tarim Osnof Guirruço, cuja vinda ao mundo trouxe mais luz, sabedoria e alegria para a minha vida.

À minha mãe, Irene Armindo Nhar, que incansavelmente, dia e noite, houvesse chuva ou sol, batalhou para eu e meus irmãos pudéssemos alcançar, ao mínimo, o nível secundário de escolaridade.

À minha supervisora e docente, pelo tempo, paciência e orientação dada de forma sábia e simpática no decorrer da minha formação e elaboração da presente pesquisa.

Ao meu amigo e chefe, José Macoola Cossa, pela amizade e ensinamentos profissionais e de vida, que de certo levarei pelo resto da vida.

Aos meus familiares, sempre presentes e de alguma forma contribuíram para a realização da minha formação.

Aos meus colegas de faculdade e docentes.

## RESUMO

A presente pesquisa denominada “Desenvolvimento e Implementação de Sistema de Facturação: Caso da empresa COSYS, Lda” foi desenvolvida na empresa de tecnologias, COSYS, e tem por objectivo conceber e implementar uma solução informática para digitalizar o processo de emissão e controlo de facturas. Foi usada uma abordagem literária que implicava a análise da documentação da empresa. Para início do desenvolvimento do sistema, optou-se por modelá-lo usando o Unified Modeling Language, que especifica as linhas claras de projecto e codificação a seguir no desenvolvimento de softwares.

Em termos de organização do trabalho, primeiro será apresentada a problemática da questão e a metodologia proposta para a resolução da mesma. No capítulo a seguir, procura-se trazer toda a documentação literária que suporta o norteou o autor da mesma. No terceiro capítulo, apresenta-se os resultados da pesquisa, para no quarto capítulo serem espelhadas as lições tiradas da elaboração da mesma e os passos que sugerem ser subsequentes (como recomendações no quinto capítulo).

**Palavras-chave:** facturação, software, ágil.

## ABSTRACT

The present research called “Development and Implementation of Invoicing System: Case of the company COSYS, Lda” was developed in the technology company, COSYS, and aims to design and implement an IT solution to digitize the process of issuing and controlling invoices. A literary approach was used which involved analyzing the company's documentation. To start developing the system, we chose to model it using the Unified Modeling Language, which specifies the clear lines of design and coding to be followed in software development.

In terms of work organization, the issue of the issue and the proposed methodology for solving it will first be presented. In the following chapter, an attempt is made to bring all the literary documentation that supports its author's orientation. In the third chapter, the results of the research are presented, in order to reflect in the fourth chapter the lessons learned from its elaboration and the steps that suggest to be subsequent (such as recommendations in the fifth chapter).

**Keywords:** *invoicing, software, agile*

## **CAPÍTULO I - INTRODUÇÃO**

Com o surgimento do computador e o avanço da tecnologia, no geral, a gestão das instituições tende cada vez mais a abraçar estas novas tendências mundiais.

Estas tecnologias mostram-se benéficas pois conseguem responder as necessidades de várias instituições e indivíduos, de forma general ou específica, conforme a sua arquitectura.

Todo e qualquer sistema é desenvolvido para responder a uma ou mais necessidades específicas. A Informática veio revolucionar a gestão das instituições no que concerne à organização e troca de informação, diária ou indefinida. Portanto, algumas questões relativas à “saúde” da própria organização só podem ser visualizadas após um certo período de tempo e referem-se a períodos de tempo definidos. Nem sempre é um processo linear e simples, mas com o advento das TIC's, mais precisamente os sistemas personalizados, a organização tem a possibilidade de minimizar as dificuldades sentidas por esta no decurso das suas actividades.

O presente trabalho é fruto de uma investigação científica que engloba as áreas de gestão de negócios e informática, e está dividido em 3 (três) capítulos, citando:

Capítulo I - Introdução: onde se apresentam os objectivos a alcançar através deste projecto, as razões que orientaram a produção do trabalho, a metodologia usada e a descrição da empresa onde se desenvolveu a pesquisa;

Capítulo 2 - Enquadramento Teórico: onde se apresenta todo o suporte literário do projecto;

Capítulo 3 - Apresentação de Resultados: onde se apresenta a descrição do software produzido no âmbito desta pesquisa.

### **1.1. Justificativa**

O desenvolvimento de *software* é uma área da Informática que tem registado uma evolução ao longo dos anos e atraído vários indivíduos e instituições. De modo a maximizar o lucro através da comercialização da propriedade intelectual ou ainda pela venda de licenças de uso dos produtos de *software*, estas entidades procuram desenvolver produtos comerciais de tipo genérico, ou seja, um *software* padronizado e quase imutável obrigando, desta forma, que a instituição adapte-se ao *software* e não o contrário, resultando na diferença entre os procedimentos da instituição e os do *software* em causa. Além das questões levantadas, este tipo de *software* não propicia a evolução do produto a nível interno institucional.

A falta ou quase total inexistência de um produto de *software* de origem moçambicana *open-source* que visa atender os requisitos das instituições no que concerne à gestão de processos de facturação e o grande interesse do autor deste trabalho despertaram a vontade de providenciar uma solução informática personalizada e que respeitasse ou estivesse em conformidade com a realidade ou procedimentos moçambicanos. Desta forma, no futuro, tornar-se-a possível a adaptação, correcção e evolução da solução proposta, possibilitando assim, a resposta à várias solicitações em curto espaço de tempo. Ainda, poderá haver uma minimização dos custos de manutenção da solução.

### **1.2. Problematização**

A COSYS – Consultoria e Sistemas de Informação, Lda - é uma empresa moçambicana cujas áreas de actuação são o desenvolvimento de sistemas informáticos e consultoria de gestão. Possui vários sistemas digitais que visam responder às necessidades específicas das instituições que, por meio de contratos com valor jurídico, são seus clientes.

Os contratos entre a COSYS e seus clientes pressupõem, de forma simplificada, a prestação de serviços pela COSYS e o pagamento de valores monetários, pelos clientes, definidos em contrato.

Contudo, a COSYS não possui um sistema informático que lhe permita gerir o registo de dívidas e pagamentos pelos serviços prestados aos seus clientes.

Actualmente, as facturas são emitidas pelos sectores de administração e secretaria recorrendo à planilhas de Excel. Tal prática implica que, no caso de algum funcionário necessitar emitir uma factura, deverá primeiro consultar o último registo no livro de facturas manuais.

Após de uma emissão da factura, deve-se cadastrar alguns dados da mesma em outra planilha Excel, categorizando a factura por cliente e projecto respectivos. Esta planilha serve de base para a consulta do estado da factura e produção de relatórios. Sendo um processo manual, a falta de registo e/ou actualização desta planilha pode significar a compilação de informação errónea. Ainda importa referir que, sendo um ficheiro acessível apenas para o nível de gestão, a informação nela contida deve ser fidedigna e altamente protegida de adulterações indevidas.

Nota-se uma enorme redundância de dados por falta de uma base de dados centralizada e, ainda por este motivo, a informação de uso corrente não é dada em tempo real, necessitando sempre de actualização e conferência pelos funcionários.

Devido à descentralização da base de dados em ficheiros Excel, torna-se complicado manter backups efectivos da informação ligada à gestão de facturas nesta instituição, além de não ser possível recuperar ou continuar a execução de uma tarefa no caso de corte no fornecimento de corrente eléctrica.

A partir do descrito acima, procura-se criar uma solução informática que visa automatizar os processos de gestão de facturas e reduzir o esforço despendido pelos funcionários. O nome escolhido para a mesma é **coFactura**.

### **1.3.Objectivos**

#### **1.3.1. Objectivo Geral**

- Conceber e implementar um sistema de gestão de facturação com vista a automatizar a emissão de facturas e recibos.

### **1.3.2. Objectivos Específicos**

- Analisar a eficácia do desenvolvimento de *softwares* recorrendo à metodologia ágil;
- Analisar o impacto dos sistemas de informação na simplificação dos processos administrativos;
- Identificar as principais necessidades da instituição correspondentes à gestão das dívidas e pagamentos pelos serviços aos seus clientes;
- Propor a flexibilização do processo de produção de relatórios financeiros e colmatação de erros por parte dos funcionários durante esse processo.

### **1.4. Questões de Pesquisa**

- Quais são as vantagens das metodologias de desenvolvimento ágil no processo de concepção até a implementação do *software*?
- Como o coFactura responderá às necessidades específicas da COSYS a nível da gestão de facturas electrónicas?
- Qual é a garantia de autenticação e autorização no sistema, ou seja, de acesso aos recursos do sistema somente aos usuários credenciados?
- Quais são os mecanismos a serem implementados para assegurar a disponibilidade de dados?
- De que forma o sistema coFactura ajudará a melhorar a programação financeira da COSYS?
- Que ganhos haverá ao se implementar esta nova forma de procedimento?

### **1.5. Hipóteses**

*H0* – O desenvolvimento ágil acelera o desenvolvimento de *software* pois permite que este seja criado e implementado em fases.

*H1* – As etapas principais são a recolha de dados e a refinação de requisitos, pois estes norteiam todo o processo de desenvolvimento do *software*.

*H2* – A implementação do conceito de permissões por perfil de usuário e o uso do SSL nas conexões entre o servidor e o cliente garantem a segurança do sistema e privacidade dos usuários.

*H3* – Garantia de disponibilidade de dados e não-perda dos mesmos através da replicação.

*H4* – Uso automático de funções embutidas no sistema, a nível da base de dados, para o cálculo de impostos, não sendo, portanto, a manipulação directa destes dados.

*H5* – Redução significativa do tempo despendido na elaboração de relatórios financeiros.

### **1.6.Importância do Tema**

A tecnologia da informação inovou o mundo dos negócios. Os processos empresariais precisam ser dotados de confiabilidade, versatilidade, eficiência e eficácia.

A tecnologia de informação é utilizada para melhorar o desempenho das actividades da empresa, e por consequência, apoiar a reengenharia dos processos empresariais. A busca por estas melhorias é que leva os gestores a criarem soluções informáticas, unindo partes e coordenando-as de forma a que, trabalhando em conjunto, os objectivos da organização sejam alcançados.

O'Brien (2002, pág. 250), descreve que um sistema de informações gerenciais gera informações que apoiam as necessidades dos administradores, que recorrem a estas, como subsídio às suas actividades de tomada de decisão. Portanto, os processos operacionais tendem a tornar-se mais eficientes, e os processos gerenciais da empresa mais eficazes. Com essas melhorias nos processos empresariais a empresa pode reduzir custos, melhorar a qualidade e o atendimento ao cliente e criar novos produtos e serviços.

Garantindo uma padronização e o acompanhamento dos processos em toda a empresa, trazendo práticas de mercado em forma de modelos a serem implementados e garantindo o alinhamento dos objetivos estratégicos da empresa em todos os níveis da organização, certamente os resultados aparecerão.

Conjugando as linhas anteriores, fica claro que há uma necessidade urgente de controlar, sistemática e integralmente, o seu fluxo financeiro que advêm dos serviços prestados a terceiros, sem prejuízo da qualidade. Pois as empresas, tal como a COSYS, existem para ganhar dinheiro.

### **1.7. Metodologia do Projecto**

Esta pesquisa é do tipo aplicada. De acordo com Barros e Lehfeld (2000, pág. 78), “a pesquisa aplicada tem como motivação a necessidade de produzir conhecimento para aplicação de seus resultados, com o objectivo de contribuir para fins práticos, visando à solução mais ou menos imediata do problema encontrado na realidade”. Appolinário (2004, pág. 152) salienta que pesquisas aplicadas têm o objetivo de “resolver problemas ou necessidades concretas e imediatas.”

“As pesquisas aplicadas dependem de dados que podem ser colectados de formas diferenciadas, tais como pesquisas em laboratórios, pesquisa de campo, entrevistas, gravações em áudio e/ou vídeo, diários, questionários, formulários, análise de documentos, etc”, (OLIVEIRA, 2007, pág. 69). Ao contrário da pesquisa teórica, investigações de natureza aplicada apresentam complexidades metodológicas e éticas muito mais complexas. Na maioria dos casos, as pesquisas aplicadas exigem e partem de estudos teóricos.

#### **1.7.1. Técnicas de Recolha de Dados**

Moresi (2003, pág. 64) define técnica de recolha de dados como "o conjunto de processos e instrumentos elaborados para garantir o registro das informações, o controle e a análise dos dados". Nos próximos parágrafos, serão definidas algumas que se presumem de interesse nesta pesquisa.

A observação não participante, também conhecida como passiva, é definida como aquela em que quem observa apenas se limita a fazê-lo de forma neutra, ou seja, permanecendo alheio aos dados colhidos, posicionando-se do lado de fora e se mantendo como mero expectador. A observação directa é aquela em que a situação está a ser observada enquanto ocorre.

Questionário é definido por Marconi & Lakatos (2003, pág. 201), como “um instrumento de recolha de dados constituído por uma série ordenada de perguntas, que devem ser respondidas por escrito e sem a presença do entrevistador”.

A revisão da documentação visa complementar o trabalho de observação no campo e/ou de entrevista através da recolha de informação e da posterior análise de aspectos documentados que foram gerados no âmbito das actividades relacionadas com o problema em estudo, tais como leis, regulamentos, contractos, correspondência, memorandos ou registos de rotina. Este tipo de documentos podem ser uma fonte interessante de informação sobre as actividades realizadas e os processos que aconteceram, e podem vir a gerar ideias para novas questões que podem ser retomadas através de observação ou de entrevista.

A recolha de dados para o sucesso será realizada através da observação directa e não participante das actividades realizadas, revisão da documentação da área e questionário com perguntas abertas aplicado à direcção da COSYS.

### **1.7.2. Metodologia de Desenvolvimento da Aplicação**

Para garantir os objectivos de desenvolvimento da aplicação proposta, será usado o *Rational Unified Process* – RUP – que é um processo de engenharia de software que fornece uma abordagem disciplinada para assumir tarefas e responsabilidades dentro de uma organização de desenvolvimento, cujo objetivo é assegurar a produção de *software* de alta qualidade dentro de prazos e orçamentos previsíveis (KRUCHTEN, 2003, pág. 14)..

### **1.7.3. Campo de Estudo**

A pesquisa será desenvolvida na COSYS - Consultoria e Sistemas de Informação, Lda, que é uma empresa moçambicana que tem como principais áreas de actuação a consultoria, desenvolvimento, implementação, manutenção e treinamento em Sistemas de Informação, Gestão e Contabilidade,

fundada em 2012. Seus principais clientes são instituições de ensino públicas e privadas que operam no território moçambicano. Tem como sede o bairro Mozal, comunidade de Djuba, Quarteirão 1, Talhão 54, com NUIT 400403333, distrito de Boane, província de Maputo.

## **CAPÍTULO II – ENQUADRAMENTO TEÓRICO**

### **2.1. Facturação**

Há séculos que a humanidade percebeu a não-autossuficiência de cada indivíduo e/ou tribo/povo. Essa constatação desempenhou um papel importante no estabelecimento de trocas comerciais como parte essencial da vida quotidiana. Nos tempos primordiais, este comércio era baseado em trocas directas de produtos, como por exemplo: sal por peixe, milho por algodão, pastagens por ferragens.

Durante a evolução da ciência e da técnica, surgiram as primeiras moedas, instrumentos que quantificariam o valor de um produto fornecido ou serviço prestado. Percebeu-se, ainda, que havia necessidade de catalogar os bens comercializados. Desta forma, a comercialização de um bem ou serviço devia ser comprovado, sendo que tanto o fornecedor assim como o cliente tivessem um comprovativo da transação. A este mecanismo, foi dado o nome de factura.

Segundo o *website* [vendus.com](http://vendus.com) (2021) “uma factura é um documento que deve ser emitido sempre que se adquire um bem ou serviço sujeito a IVA, mesmo que esta não seja solicitada pelo cliente. Na prática, o acto de comprar um produto ou de contratar um determinado serviço deve dar origem à emissão de uma factura. Por exemplo, na compra de um computador ou na contratação de serviços de marketing, cabe ao fornecedor ou prestador de serviços emitir um documento comprovativo da compra.”

Pode-se definir facturação como sendo o valor resultante após a emissão de uma ou mais facturas pela empresa.

#### **2.1.1. Tipos de Factura**

O *website* “vendus.com” (2021) classifica as facturas em:

##### **✓ Factura Recibo**

➤ É o tipo de documento que se deve emitir quando a data da factura e do pagamento coincidem (pronto pagamento). Este documento agrega a factura e o recibo, não havendo necessidade de existir um recibo separado.

✓ **Fatura Proforma**

➤ É um documento informativo frequentemente utilizado na cotação e transação internacional de produtos. Após aprovação do cliente pode ser convertida em factura. Embora não tenha validade fiscal, a factura pró-forma deve ser comunicada à Autoridade Tributária.

✓ **Factura Manual**

➤ É um modelo em papel pré-impressol por tipografias autorizadas pela Ministério das Finanças, cabendo ao vendedor ou prestador de serviço o preenchimento manual, usando uma caneta, dos dados da transacção.

✓ **Factura Electrónica**

➤ Documento emitido por software de facturação certificado pela Autoridade Tributária, podendo ser impresso ou enviado por email. A esta factura está associada uma assinatura digital que permite validar a autenticidade do documento.

### **2.1.2. Elementos de uma Factura**

Segundo a Autoridade Tributária de Moçambique, uma factura ou equivalente (documento fiscalmente aceite) emitida em território nacional só é válida se incluir os seguintes componentes:

- ✓ Deve ser emitida em língua portuguesa e em Meticais, (n.º 5, artigo 106 da Lei n.º 2/2006, de 22 de Março);
- ✓ Data, (n.º 5, artigo 27 do CIVA);

- ✓ Numeração sequencial impressa em tipografia autorizada pelo Ministério das Finanças, (n.º 5 e 7, artigo 27 CIVA) ou por computador;
- ✓ Indicação dos elementos identificativos da tipografia, nomeadamente a designação social, sede e NUIT, bem como a autorização prevista no n.º 7, artigo 27 do CIVA;
- ✓ Identificação completa (Nome, NUIT e domicílio do adquirente), (alínea a) n.º5, artigo 27 do CIVA);
- ✓ Quantidade e denominação usual dos bens/serviços, (alínea b) n.º 5, artigo 27 do CIVA);
- ✓ Indicação separada das embalagens não transaccionadas, (alínea b) n.º 5, artigo 27 do CIVA);
- ✓ Preço líquido de imposto e outros elementos incluídos no valor tributável, (alínea. c) n.º 5 artigo 27 do CIVA);
- ✓ Taxa e montante do imposto devido, (alínea d) n.º 5, artigo 27 do CIVA); e
- ✓ Motivo justificativo da não aplicação do Imposto, (alínea e) n.º 5, artigo 27 do CIVA).

Com a aprovação da Lei de Transacções Electrónicas em Moçambique, as facturas electrónicas passaram a ter valor fiscal, contribuindo para flexibilização da informatização das empresas. Para tal, as empresas tem recorrido cada vez mais ao uso de TIC's para controlar a sua facturação.

## **2.2. Tecnologias de Informação e Comunicação**

Segundo o *website* infoescola.com (2020) “Tecnologias de Informação e Comunicação são o conjunto de recursos tecnológicos, utilizados de forma integrada, com um objetivo comum. As TIC's são utilizadas das mais diversas formas, na indústria (no processo de automação), no

comércio (gestão e publicidade), no sector de investimentos (informação simultânea, comunicação imediata) e na educação (no processo de ensino-aprendizagem, na educação a distância)”.

A expressão “TIC” foi primeiro usada em 1997 por Dennis Stevenson, do governo britânico e promovida pela documentação do Novo Currículo Britânico em 2000.

Pode-se dizer que a principal responsável pelo crescimento e potencialização da utilização das TIC’s em diversos campos foi a popularização da *internet*.

Como a comunicação é uma necessidade e algo que está presente na vida do ser humano desde os tempos mais remotos, trocar informações, registrar factos, expressar ideias e emoções são factores que contribuíram para a evolução das formas de se comunicar. Assim, com o passar do tempo, o homem aperfeiçoou sua capacidade de se relacionar.

Nesse sentido, conforme as necessidades surgiram, o homem lançou mão de sua capacidade racional para desenvolver novas tecnologias e mecanismos para a comunicação. Conceitua-se tecnologia como tudo aquilo que leva alguém a evoluir, a melhorar ou a simplificar. Em suma, todo processo de aperfeiçoamento. A humanidade já passou por diversas fases de evoluções tecnológicas, porém um equívoco comum quando se pensa em tecnologia é se remeter às novidades de última geração.

Tratando-se de informação e comunicação, as possibilidades tecnológicas surgiram como uma alternativa da era moderna, facilitando a educação através da inclusão digital, com a inserção de computadores nas escolas, facilitando e aperfeiçoando o uso da tecnologia pelos alunos, o acesso a informações e a realização de múltiplas tarefas em todas as dimensões da vida humana, além de capacitar os professores por meio da criação de redes e comunidades virtuais.

“Os computadores são grandes responsáveis por esse processo. Os Sistemas de Informação nas empresas requerem estudos quanto à sua importância na abordagem gerencial e estratégica dos mesmos, juntamente com a análise do papel estratégico da informação e dos sistemas na empresa” (KROENKE, 1992).

### **2.3. Arquitectura dos Sistemas Digitas de Informação e Comunicação**

Ao longo dos anos, o mundo empresarial tem apostado intensivamente no desenvolvimento e fortalecimento do uso de ferramentas digitais como mecanismos dinâmicos de comunicação. Mais importante do que a comunicação, é o registo e acesso à informação que faz parte das actividades diárias(e não só) das instituições. Para atingir estes objectivos, tem sido desenvolvidos sistemas informáticos, que podem ser classificados na sua acessibilidade em duas principais arquitecturas: *desktop* e *web*.

A aplicação *desktop* é instalada num computador para ser usada no mesmo sem necessidade de acesso à internet, podendo-se no entanto ser usado por vários computadores numa rede local. As maiores desvantagens desta arquitectura são as grandes dificuldades de manutenção, actualização e comercialização.

A aplicação *web* é instalada em apenas um computador (servidor) e podendo ser acedida através de vários computadores e outros dispositivos informáticos através de um navegador, dentro ou fora da rede em que este servidor está localizado. Permite que haja alto tráfego e acessos concorrentes sem afectar a sua performance. Para disponibilizar a aplicação, recorre-se a um software denominado *webserver*. Esta arquitectura será a base para a pesquisa aqui proposta.

Quando o objectivo é registar informação e a posterior recuperar a mesma, tanto a aplicação *desktop* quanto a *web* fazem uso de um componente informático denominado **base de dados**.

#### **2.3.1. Base de Dados**

Segundo POBLACIÓN (2006, pág. 237), “as bases de dados são compreendidas como fontes de informação electrónicas, pesquisáveis de modo interativo ou conversacional através de um computador.”

SGBD é o conjunto de programas de computador (*softwares*) responsáveis pela gestão de uma base de dados. Seu principal objectivo é retirar da aplicação-cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em bases de dados relacionais (conceito a abordar no ponto 2.3.1.2), a interface é constituída pelas API's (*Application Programming Interface*) ou *drivers* do SGBD, que executam comandos na linguagem SQL.

SQL é a linguagem de pesquisa declarativa padrão para base de dados relacional. Muitas das características originais do SQL foram inspiradas na álgebra relacional. Um dos SGBD's mais versáteis e usados actualmente no mundo é o PostgreSQL, que é um sistema de gestão de base de dados objecto-relacional baseado no POSTGRES Versão 4.2 e desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley.

Apesar de o PostgreSQL usar em sua maioria a linguagem SQL, foi desenvolvida uma extensão desta linguagem, chamada PLPGSQL. A PLPGSQL ou PL/pgSQL é uma linguagem estrutural que funciona como uma extensão da SQL. Esta linguagem tem por objectivo auxiliar as tarefas de programação no PostgreSQL. Ela incorpora à SQL características procedurais, como os benefícios e facilidades de controle de fluxo de programas que as melhores linguagens possuem, como, por exemplo loops estruturados (*for*, *while*) e controle de decisão (*if then else*).

Usar a PLPGSQL traz vantagens, das quais podemos citar:

- Suporte a extensões de linguagem;
- Estrutura de selecção;
- Estrutura de loop;
- Combinação com SQL declarativo;
- Combinação com transacções;
- Tratamento de excepções;
- Suporte a escopo de variáveis;

- Suporte aos tipos primitivos, complexos e domínios ( definidos pelo usuário).

#### 2.3.1.1. Função da base de dados

- Promover o acesso à informação;
- Fornecer informações atualizadas, precisas e confiáveis;
- Atender às necessidades do público alvo;
- Fornecer mecanismos eficientes de recuperação.

#### 2.3.1.2. Modelos de base de dados

Segundo o *website* “<https://pt.wikipedia.org>” (2020), os modelos de base de dados actualmente reconhecidos são:

- **Modelo Plano ou Tabular:** consiste de matrizes simples, bidimensionais, compostas por elementos de dados: inteiros, números reais, etc. Este modelo plano é a base das planilhas eletrônicas;
- **Modelo em Rede:** permite que várias tabelas sejam usadas simultaneamente através do uso de apontadores (ou referências). Algumas colunas contêm apontadores para outras tabelas ao invés de dados. Assim, as tabelas são ligadas por referências, o que pode ser visto como uma rede;
- **Modelo Hierárquico:** é uma variação particular do modelo em rede, limita as relações a uma estrutura semelhante a uma árvore (hierarquia - tronco, galhos), ao invés do modelo mais geral direcionado por grafos;
- **Base de Dados Relacionais:** consistem, principalmente de três componentes: uma coleção de estruturas de dados, nomeadamente relações, ou informalmente tabelas; uma coleção dos operadores, a álgebra e o cálculo relacionais; e uma coleção de restrições da integridade, definindo o conjunto consistente de estados de base de dados e de alterações

de estados. As restrições de integridade podem ser de quatro tipos: domínio (também conhecidas como *type*), atributo, relvar (variável relacional) e restrições de base de dados.

### 2.3.2. WebServer

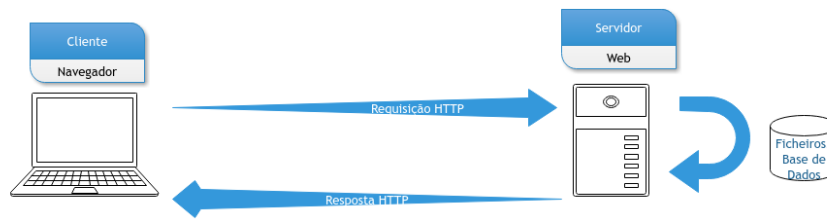
*WebServer* é um *software* responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas *web*, tais como documentos HTML com objectos embutidos (imagens, etc.). Pode-se usar este termo para indicar um computador que executa um programa que provê a funcionalidade descrita acima.

Dentre os *webservers* mais usados no mundo, destacam-se: Apache, Nginx, Apache Tomcat e Microsoft's Internet Information Services (IIS) Windows Server.

#### 2.3.2.1. Princípios de Funcionamento de um *WebServer*

Um *webserver* inclui diversos componentes que controlam como os usuários acessam os arquivos hospedados, sendo destes, no mínimo, um servidor HTTP, que é um software que compreende URL's (endereços web) e HTTP (o protocolo que seu navegador utiliza para visualizar páginas web).

Em um nível mais básico, o navegador fará uma requisição utilizando o protocolo HTTP sempre que necessitar de um um arquivo hospedado em um servidor web. Quando a requisição alcançar o servidor web correcto (*hardware*), o servidor HTTP (*software*) enviará o documento requerido, também via HTTP, conforme mostra a figura abaixo:



**Figura 1:** Modelo Conceitual do Webservice

Fonte: <https://study-ccna.com/> (adaptado)

Esta chamada das páginas pode ser feita através de *links* presentes numa determinada página, sendo a nova página carregada na totalidade ou em fragmentos, usando bibliotecas de execução de *scripts* no navegador, como é o caso do *jQuery*.

O *jQuery* é uma biblioteca leve, rápida e cheia de recursos baseada no Javascript. Facilita a manipulação de eventos, animações, elementos *HTML* e utilização de *Ajax*.

O *jQuery* foi lançado oficialmente em 2006 e possui código aberto. A biblioteca também oferece a possibilidade de criação de plugins sobre ela. Através do *jQuery*, é possível desenvolver aplicações web de alta complexidade.

Dentre os pontos que caracterizam o *jQuery*, podemos destacar:

- É bem leve (a versão compactada e minificada da biblioteca possui apenas 32kB);
- Suporte a diversos navegadores web
  - ✓ O *jQuery* funciona em todos os navegadores mais utilizados no mundo, entre eles: Google Chrome, Firefox, Edge, Safari, Internet Explorer, Android, iOS;
- Suporte a CSS3
  - ✓ Por consequência da integração com HTML, também suporta selectores CSS3 para encontrar elementos, bem como na manipulação de propriedades de estilo.

Tomando como exemplo o *webserver* chamado Apache, e no âmbito da pesquisa aqui apresentada, foram desenvolvidos módulos que permitem o processamento dinâmico das páginas, ou seja, usar a programação para a criação de *scripts* que permitam a gravação, recuperação e eliminação da informação, tanto gravada em disco assim como em base de dados. Tal é o caso do **PHP**.

### 2.3.2.1.1. PHP

Segundo o *website* “<https://monografias.brasilecola.uol.com.br>” (2020), o PHP foi criado durante o Outono de 1994 por Rasmus Lerdord. É uma linguagem de *script* interpretada que permite criar *websites* dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros da *url* e *links*. O que distingue o PHP e outras linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas *html* puro. Desta forma, é possível interagir com bases de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente.

O PHP tem como características principais:

- Linguagem de fácil aprendizado;
- Tem suporte a um grande número de base de dados como, por exemplo, dBase, Interbase, mSQL, MySQL, Oracle, Sybase e PostgreSQL;
- Tem suporte a outros serviços através de protocolos como HTTP, IMAP, SNMP, NNTP e POP3;
- É multi-plataforma, tendo suporte aos sistemas operativos mais utilizados no mercado;
- *Open-source*;
- Não precisa ser compilado.

O PHP é uma linguagem de alta *performance*, sendo por isso usada em muitos sistemas e páginas *web* a nível mundial. Porém, desenvolver uma aplicação usando um certo SGBD + PHP não

fornece grande facilidade para a migração da mesma aplicação para outro SGBD. Como forma de abstracção sobre o SGBD usado, criou-se uma poderosa classe chamada de PDO.

### **2.3.2.1.2. PDO**

A classe PDO - PHP Data Objects - é uma classe desenvolvida especificamente para trabalhar com procedimentos relacionados a base de dados.

O uso do PDO permite a abstracção de qual banco utilizamos e a segurança extra que esta classe nos oferece. Isto significa que o PDO oferece recursos suficientes para trabalhar implementando todo o *software* sem preocupação com o tipo de base de dados a ser usado. Uma mudança posterior no tipo de base de dados não trará grandes problemas ao *software*.

#### **2.3.2.1.2.1. Características do PDO**

➤ **Flexibilidade**

✓ O PDO carrega o driver específico da base de dados no tempo de execução, não é preciso reconfigurar o PHP sempre que um banco de dados diferente for usado.

➤ **Alto Desempenho**

✓ O PDO está escrito em C e compilado no PHP, o que lhe garante um aumento considerável no desempenho em relação a soluções escritas em PHP.

➤ **Consistência de código**

✓ No PDO não existe a inconsistência de código, pois é oferecida apenas uma interface unificada que é está disponível para qualquer base de dados.

➤ **Orientação à objetos**

✓ Possui recursos de orientação de objetos, o que resulta em uma comunicação mais poderosa e eficiente com a base de dados.

## **2.4. Desenvolvimento de Sistemas Digitais de Informação e Comunicação**

### **Ferramentas CASE**

CASE é uma classificação que abrange todas ferramentas baseadas em computadores que auxiliam actividades de engenharia de *software*, desde análise de requisitos e modelagem até programação e testes. Podem ser consideradas como ferramentas automatizadas que tem como objetivo auxiliar o desenvolvedor de sistemas em uma ou várias etapas do ciclo de desenvolvimento de *software*.

Kotonya e Sommerville (1998, pág. 39) classificam essas ferramentas em dois tipos:

- ✓ **Ferramentas de modelagem e validação:** apoiam o desenvolvimento de modelos de sistema, que podem ser usados para especificar o sistema e a verificação desses modelos quanto à consistência e completude. São exemplos os *softwares* de modelagem Astah, Poseidon e ArgoUML;
- ✓ **Ferramentas de gestão:** auxiliam na gestão da base de dados dos requisitos e apoiam a gestão de mudanças desses requisitos. São exemplos o RequisitePro, da IBM/Rational ou a Enterprise Architect, da Sparx Systems.

#### **2.4.1. Ciclo de vida de Um Sistema Informático**

O ciclo de vida é a estrutura contendo processos, actividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de *software*, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.

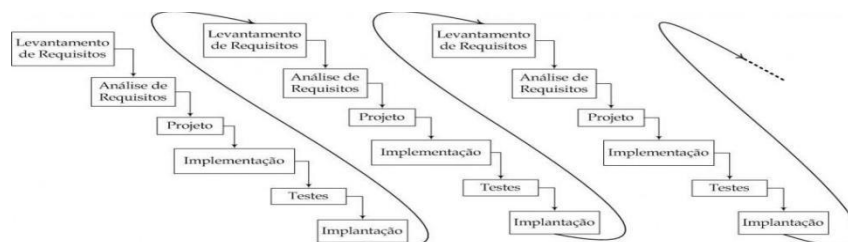
O modelo de ciclo de vida é a primeira escolha a ser feita no processo de *software*. A partir desta escolha, definir-se-á desde a maneira mais adequada de obter as necessidades do cliente, até quando e como o cliente receberá sua primeira versão operacional do sistema.

### 2.4.1.1. Modelo Iterativo e Incremental

O modelo iterativo e incremental é um processo de desenvolvimento de *software* criado em resposta às fraquezas do modelo em cascata, o mais tradicional. Os dois padrões mais conhecidos de sistemas iterativos de desenvolvimento são o RUP (Processo Unificado da Rational) e o Desenvolvimento ágil de *software*. Por isso o desenvolvimento iterativo e incremental é também uma parte essencial da Programação Extrema e outros.

Desenvolvimento Incremental é uma estratégia de planejamento estagiado em que várias partes do sistema são desenvolvidas em paralelo, e integradas quando completas. Não implica, requer ou pressupõe desenvolvimento iterativo ou em cascata. Neste modelo, o tempo de revisão e melhorias de partes do sistema é pré-definido. Isto não pressupõe desenvolvimento incremental, mas funciona muito bem com ele.

### 2.4.1.2. Fases do Modelo Incremental



**Figura 2:** Modelo Incremental

Fonte: website [https://wiki.sj.ifsc.edu.br/index.php/Ciclo\\_de\\_Vida\\_Iterativo\\_e\\_Incremental](https://wiki.sj.ifsc.edu.br/index.php/Ciclo_de_Vida_Iterativo_e_Incremental)

### **2.4.1.3. Vantagens do Modelo Incremental**

- Entregas parciais facilitam a identificação e correção de erros entre os componentes do *software*.
- Necessidades não especificadas nas fases iniciais podem ser desenvolvidas nos incrementos.
- Cada iteração produz um conjunto de itens utilizáveis.
- Os feedbacks de iterações anteriores podem ser usados nos próximos incrementos.
- Os incrementos podem ser desenvolvidos por menos profissionais.
- Entrega dos incrementos pode permitir o cumprimento do prazo especificado.
- Facilita a manutenção dos “módulos”.

### **2.4.1.4. Desvantagens do Modelo Incremental**

- Número de iterações não pode ser definido no início do processo.
- O fim do processo não pode ser previamente definido.
- Gestão e manutenção do sistema completo podem se tornar complexos.
- Gestão do custo é mais complexo devido ao número de iterações.

## **2.4.2. Ferramentas de Desenvolvimento de Softwares**

### **2.4.2.1. UML**

Segundo Silva e Videira (2001, pág. 117), a UML é “uma linguagem para especificação, construção, visualização e documentação de artefactos de um sistema de software. A UML disponibiliza uma forma padrão de modelagem de projectos de sistemas informáticos, incluindo

seus aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, processos de banco de dados e componentes de software reutilizáveis”. Essa linguagem se tornou, nos últimos anos, a linguagem-padrão de modelagem de *software* adotada internacionalmente pela indústria de Engenharia de Software.

A UML é composta por muitos elementos de modelo que representam as diferentes partes de um sistema de *software*. Estes elementos são usados para criar diagramas, que representam uma determinada parte, ou um ponto de vista do sistema.

Usando a UML, é possível construir os seguintes diagramas:

- **Diagrama de Caso de Uso:** mostra os actores (pessoas e/ou outros utilizadores do sistema), casos de uso (os cenários onde eles usam o sistema), e seus relacionamentos ou interações;
- **Diagrama de Classe:** mostra classes e os relacionamentos entre elas. No geral, tende-se a demonstrar a estrutura das tabelas da base de dados;
- **Diagrama de Sequência:** mostra objectos e a sequência das chamadas do método feitas para outros objectos;
- **Diagrama de Colaboração:** mostra objectos e seus relacionamentos, destacando os objectos que participam na troca de mensagens;
- **Diagrama de Estado:** mostra os estados, mudanças de estado e eventos num objecto ou uma parte do sistema;
- **Diagrama de actividade:** mostra as actividades e as mudanças de uma actividade para outra com os eventos ocorridos em alguma parte do sistema;
- **Diagrama de Componente:** mostra os componentes de programação de alto nível;
- **Diagrama de Distribuição:** mostra as instâncias dos componentes e seus relacionamentos.

#### 2.4.2.2. NetBeans

IDE é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objetivo de agilizar este processo. O NetBeans é um *software* que pertence à esta categoria, sendo um ambiente de desenvolvimento integrado gratuito e de código aberto para desenvolvedores de software nas linguagens Java, JavaScript, HTML5, PHP, C/C++, Groovy, Ruby, entre outras. O IDE é executado em muitas plataformas, como Windows, Linux, Solaris e MacOS. Além disso, é usado como base de uma série de software científico de missão crítica em grandes organizações em defesa, aeroespacial, logística e pesquisa, como Boeing, Airbus Defense and Space, NASA e NATO.

Dentre os seus recursos, podemos citar:

- Editor de código fonte integrado, rico em recursos para aplicações *web* (Servlets e JSP, JSTL, EJBs) e aplicações visuais com Swing que é uma API Java para interfaces gráficas;
- Visualizador de classes integrado ao de interfaces, que gera automaticamente o código dos componentes de forma bem organizada, facilitando assim o entendimento de programadores iniciantes;
- Suporte ao Java Enterprise Edition, plataforma de programação de computadores que faz parte da plataforma Java voltada para aplicações multicamadas, baseadas em componentes que são executados em um servidor de aplicações;
- Plugins para UML;
- Interface amigável com CVS (sistema de versões concorrentes) é um sistema de controle de versão que permite que se trabalhe com diversas versões de arquivos organizados em uma pasta e localizados local ou remotamente, mantendo-se suas versões antigas e os *logs* de quem e quando manipulou os arquivos;
- Debug apurado de aplicações e componentes;
- Autocompletar avançado;
- Suporte a bases de dados.



## **CAPÍTULO III – APRESENTAÇÃO DE RESULTADOS**

O presente capítulo tem como proposta apresentar os recursos usados e os resultados obtidos em conformidade com os objectivos dispostos na presente pesquisa.

### **3.1. Apresentação do Sistema**

A implementação das técnicas de pesquisa proposta neste projecto permitiram o desenvolvimento e implementação de uma solução de *software* web de fácil extensão, manutenção e evolução.

Como tal, o *software* foi desenvolvido em fases, conforme preconizado na metodologia RUP, totalizando três iterações (ver apêndices 18 e 19). Tal feito foi possível devido à uma definição clara e concisa das necessidades da empresa.

#### **3.1.1. Requisitos do Sistema**

Segundo PRESSMAN (2006, pág. 133), “o levantamento de requisitos (também chamado elicitación de requisitos) combina elementos de resolução de problemas, elaboração, negociação e especificação. Para encorajar uma abordagem colaborativa e orientada às equipes em relação ao levantamento de requisitos, os interessados trabalham juntos para identificar o problema, propor elementos da solução, negociar diferentes abordagens e especificar um conjunto preliminar de requisitos da solução”, desta forma clarificando o que se pretende ao final do projecto. De forma geral, os requisitos podem ser classificados em:

- **Funcionais:** referem-se sobre o que o sistema deve fazer, ou seja, suas funções e informações
- **Não-funcionais:** referem-se aos critérios que qualificam os requisitos funcionais, ou seja, os requisitos de performance, usabilidade, confiabilidade, robustez, requisitos de entrega, implementação, etc.

## Requisitos Funcionais

- Gerir<sup>1</sup> Anos Efectivos
- Gerir Perfis dos Usuários
- Gerir Usuários
- Gerir Emolumentos
- Gerir Bancos
- Gerir Contas Bancárias
- Gerir Clientes
- Gerir Facturas
- Gerir Recibos
- Emissão de Relatórios:

## Requisitos Não-Funcionais

- **Software: Ambiente de produção**

*Tabela 1: Requisitos Não-Funcionais Software*

Tipo	Categoria	Nome e Versão
Operativo		Linux Ubuntu Server 14.04 LTS
Aplicação	WebServer	Apache 2.4.4
		PHP 5.5.5
Aplicação	Base de Dados	PostgreSQL 10.10

*Fonte: autor*

---

<sup>1</sup> No âmbito de requisitos funcionais desta pesquisa, “gerir” engloba as acções de registo, listagem, actualização e eliminação (CRUD)

➤ **Software: Ambiente de desenvolvimento**

*Tabela 2: Requisitos Não-Funcionais Software 2*

<b>Tipo</b>	<b>Categoria</b>	<b>Nome e Versão Usados</b>
Operativo		Linux Mint Cinnamon 19.1 (64bits)
Aplicação	<i>WebServer</i>	Apache 2.4
Aplicação		PHP 5.6
Aplicação	Base de Dados (SGBD)	PostgreSQL 10.10
Aplicação	Desenvolvimento (IDE)	NetBeans 10
Aplicação	Desenvolvimento (SGBD)	pgAdmin IV
Aplicação	Base de Dados (Modelagem)	StarUML v2beta10
Aplicação	Base de Dados (Modelagem)	Navicat Data Modeler 1.0.12

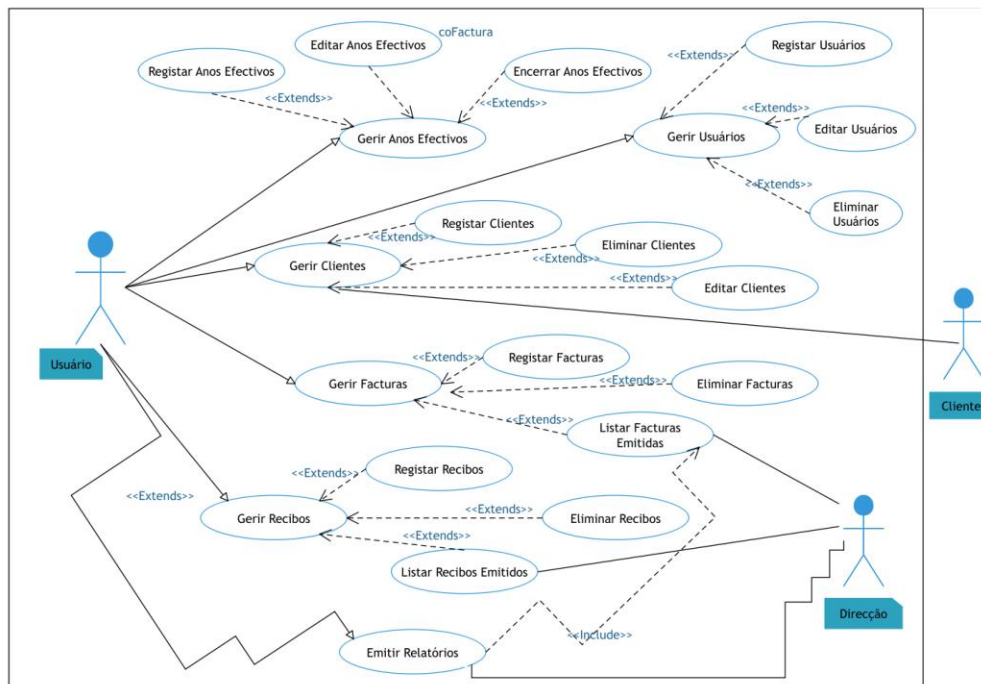
*Fonte: autor*

➤ **Hardware: Ambiente de produção**

- ✓ Um servidor com as especificações mínimas:
  - a. Disco rígido de 250GB SSD
  - b. Memória RAM de 4GB DDR3
  - c. Plataforma Dual Core com CPU Clock de 1.8GHz

### **3.2. Modelagem do Sistema**

#### **3.2.1. Diagrama dos Principais Casos de Uso**



**Figura 3:** Diagrama de Casos de Uso

Fonte: autor

### 3.2.1.1 Descrição dos Principais Casos de Uso

*Tabela 3: Descrição dos Casos de Uso*

ID	Nome	Autores	Descrição do Procedimento
R1	Gerir Usuários	Usuário	Consiste em listar os usuários do sistema
R2	Gerir Anos Efectivos	Usuário	Consiste em visualizar/listar os anos efectivos
R3	Gerir Clientes	Usuário	Consiste em visualizar/listar os clientes da empresa
R4	Gerir Facturas	Usuário	Consiste em visualizar/listar as facturas
R5	Gerir Recibos	Usuário	Consiste em visualizar/listar os recibos emitidos
R6	Emitir Relatórios	Usuário	Consiste em extrair relatórios relativos às facturas e recibos

Fonte: autor

### 3.2.2. Diagrama de Classes

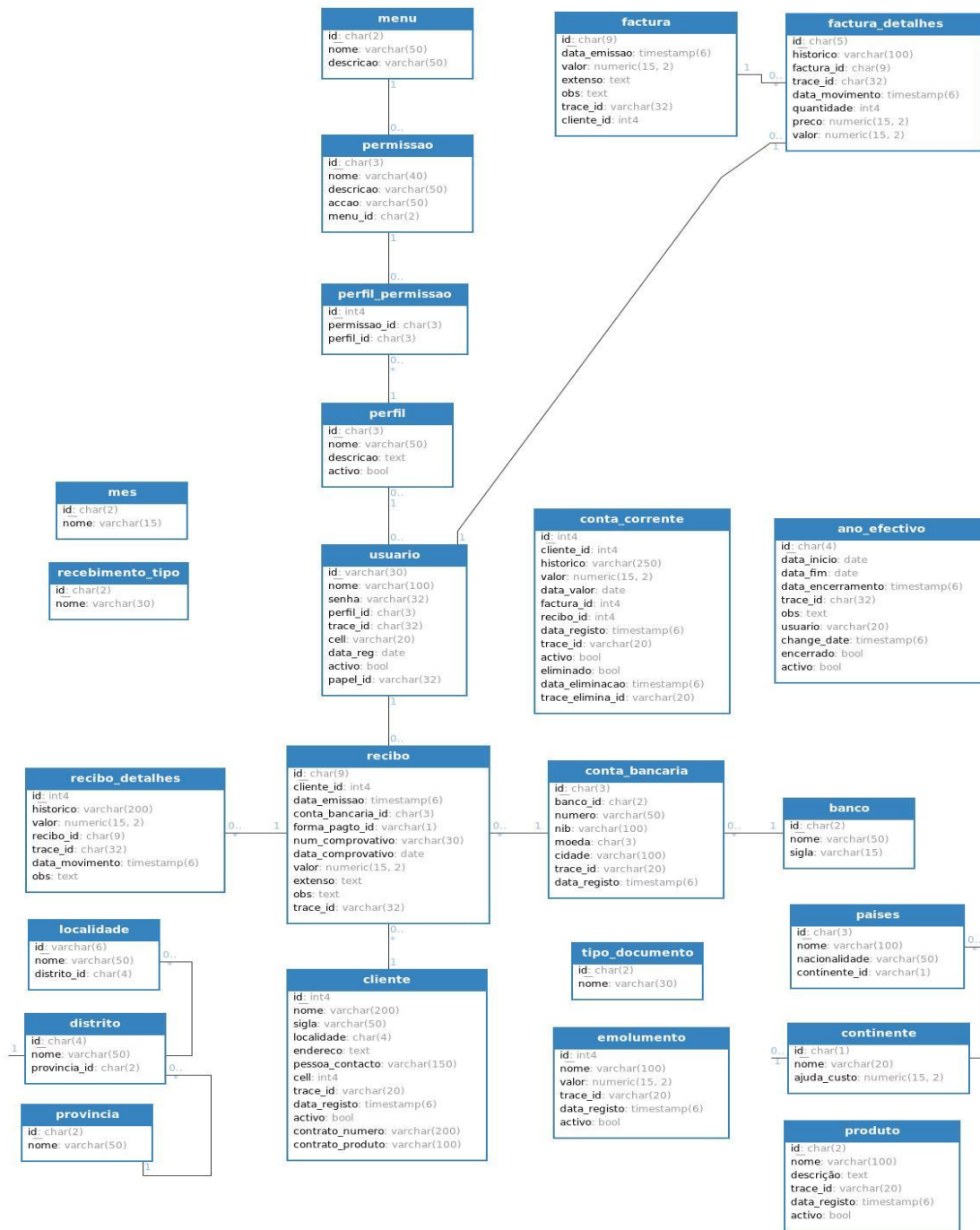


Figura 4: Diagrama de Classes

Fonte: autor

### 3.3. Funcionamento do Sistema

#### 3.3.1. Modelo de Codificação e Estruturação

Baseado nas boas práticas de desenvolvimento de aplicações e a experiência colhida ao longo dos anos na área, o autor da pesquisa optou pelo uso do modelo MVC, tendo separado a aplicação em directórios físicos, conforme o tipo a que pertence. Desta forma, à base da aplicação existem os principais:

- ✓ **view:** formulários usados no sistema (vide apêndice 20);
- ✓ **controller:** faz a conexão entre o formulário e o *model* (vide apêndice 21);
- ✓ **model:** processa todas as requisições, fazendo a ligação com a base de dados (vide apêndice 22).

#### 3.3.2. Principais Formulários

##### ✓ Login

Ao indicar a *url* em que se encontra hospedado o sistema, este é a primeira página que é apresentada, devendo o visitante indicar os dados de usuário e senha e clicar no botão “Autenticar” para ter acesso ao sistema.



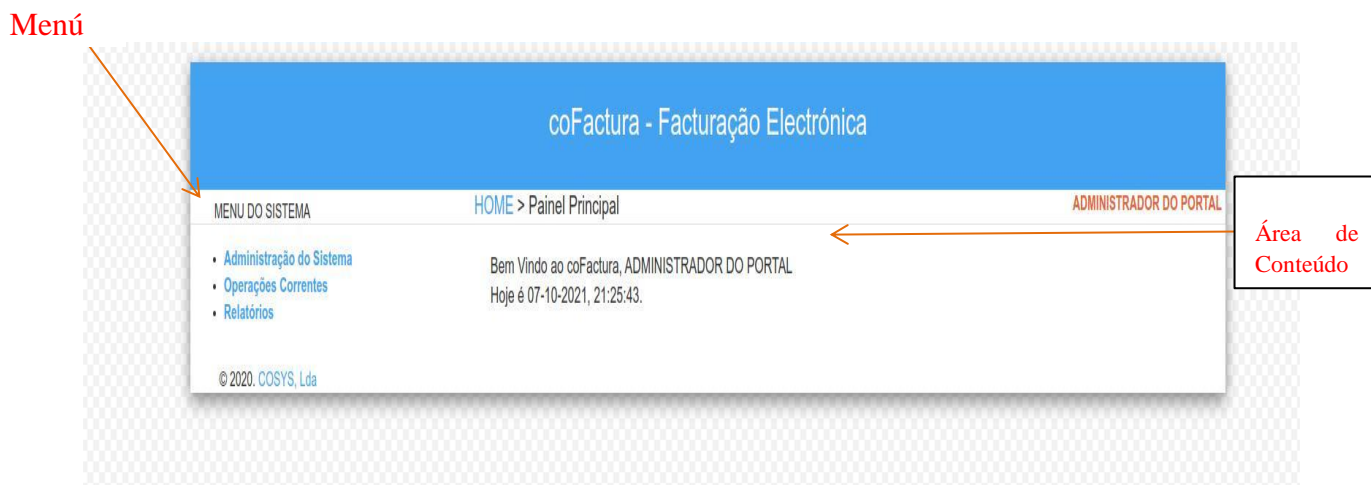
**Figura 5:** *Login*

Fonte: autor

## ✓ Painel Principal

Após a correcta autenticação, apresenta-se o painel principal que é composto de 2 partes principais:

- i. **Menú:** à esquerda, onde se localizam os *links* das funcionalidades divididas em módulos;
- ii. **Área de Conteúdo:** à direita do menú, ocupando 70% da tela. É nesta área que os formulários, mensagens e relatórios serão apresentados.



**Figura 6:** Painel Principal

Fonte: autor

## ✓ Registo de Usuário

Para cadastrar um novo usuário, deve-se clicar no módulo de **Administração do Sistema -> Gerir Usuários**. Na lista dos usuários que será apresentada, deve-se clicar em novo. Em seguida, ser-nos-à apresentado um formulário como o abaixo representado:

The image shows a web form for user registration. It is titled 'HOME > Administração > Dados de Usuário' and 'ADMINISTRADOR DO PORTAL'. The form contains the following fields: 'Login: \*' with the value 'login'; 'Senha: \*' with the value 'login'; 'Nome: \*' with the value 'Nome do Usuário'; 'Perfil: \*' with a dropdown menu showing '-- Escolher --'; 'N.º Telefone:' with the value '12346789'; and 'Activo: \*' with an unchecked checkbox. At the bottom of the form are two buttons: '✓ Guardar' and '✗ Cancelar'.

**Figura 7:** Formulário de Registo de Usuário

Fonte: autor

Vale aqui referir que:

- Na primeira autenticação, a senha será igual ao login indicado;
- O perfil atribuído dará acesso ao usuário somente às funcionalidades configuradas no mesmo.

### ✓ **Emissão de Factura**

O propósito principal do sistema é a emissão de facturas, pelo que não ficaria de lado o principal formulário, acessível através de **Operações Correntes -> Gerir Facturas -> Nova:**

HOME > Administração > Emissão de Factura ADMINISTRADOR DO PORTAL

Ano Efectivo: \* 2021

Cliente: \* Universidade São Tomás de Moçambique - 700057585

Emolumento: \* Alojamento na Cloud Montante: \* 350000

Ordem	Nome do Emolumento	Valor	Eliminar
1	Custo Anual de Uso do Sistema eDondzo	700.000,00	<input type="checkbox"/>
2	Serviço de E-mail	120.000,00	<input type="checkbox"/>
<b>Total.....</b>		<b>820.000,00</b>	

**Figura 8:** Formulário de Emissão de Factura

Fonte: autor

O primeiro passo é indicar o cliente, indicado pelo nome e NUIT. A seguir, indicar-se o emolumento a facturar, informa-se o montante e clica-se em no botão “Adicionar”. Indica-se outro emolumento e clica-se novamente no botão “Adicionar”, repetindo-se este processo quantas vezes forem necessárias. No final, gera-se a factura clicando em “Guardar”.

### 3.3.2.1. Navegação e Acções

O sistema está organizado em módulos, denominação que foi dada a uma coleção de funcionalidades co-relacionadas.

Para aceder a uma funcionalidade, dá-se um clique no módulo correspondente e a posterior na funcionalidade. Assim feito, o sistema busca o formulário correspondente através de uma chamada assíncrona - AJAX - tendo como agente de execução a ferramenta jQuery.

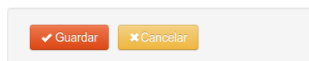
As listas/tabelas permitem que se refinem os resultados através de uma barra de pesquisa localizada na parte superior ligeiramente à direita do formulário, semelhante à figura:

Um campo de texto retangular com o rótulo "Pesquisa" à esquerda e uma borda cinza.

**Figura 9:** Campo de Pesquisa

Fonte: autor

Em cada um dos formulários, a gravação ou descarte dos dados do formulário é feita através de botões nomeados de forma intuitiva, que geralmente se apresenta da seguinte forma:

Dois botões retangulares lado a lado. O primeiro é laranja com um ícone de marca de verificação e o texto "Guardar". O segundo é amarelo com um ícone de X e o texto "Cancelar".

**Figura 10:** Camada de Botões

Fonte: autor

### 3.4. Segurança do Sistema

O coFactura usa os seguintes recursos para garantir a segurança do sistema e da informação armazenada:

- Uma encriptação de 32 bits usando o MD5 para as senhas dos usuários, sendo que estas são encriptadas antes da sua gravação na base de dados.
- Para prevenir a injeção de SQL (manipação indevida programática da base de dados), recorreu-se ao uso do *PDO* devido à sua já comprovada garantia de segurança contra este tipo de ataques.

- Todos os dados enviados pelo usuário são validados conforme o seu tipo, de forma a evitar invalidade e nulidade de dados.
- Nenhuma acção, excepto autenticação do usuário, é efectuada sem a verificação da validade da sessão do usuário e, em caso de inactividade por mais de 5 minutos, a sessão do usuário é destruída e este é obrigado a reautenticar-se.
- Recorre-se frequentemente ao uso de *Prepared Statements* de forma a garantir que as *queries* sejam executadas de forma segura. Este recurso oferece um ganho em velocidade de processamento pois as *queries* são compiladas na base de dados apenas uma vez e só aguardam os parâmetros para a sua execução.

## 4. CONCLUSÃO

Após a cautelosa revisão e análise dos resultados desta pesquisa, é possível concluir:

1. A implementação de um produto de *software* em usando a metodologia ágil é realmente benéfica pois, sendo que o projecto é levado em fases, permitirá que o *software* possa ser usado antes do prazo estabelecido do projecto. Entregando-se o produto do *software* em fase, há possibilidade de o mesmo ser usado logo no primeiro entregável enquanto se desenvolvem os outros módulos e/ou funcionalidades.
2. Nas fases de recolha de dados e refinação de requisitos, um mal-entendido pode acarretar custos financeiros elevados e de tempo investidos para consertar/reparar a secção do *software* resultante dessa falha.
3. Quando os dados e formulários do sistema ficam à disposição dos agentes-usuários do sistema, torna-se muito complicada a filtragem dos recursos. Apenas garantindo a autenticação e autorização dos usuários, aliadas aos *logs* recorrentes das suas acções no sistema, é que se pode ter total garantia de segurança.
  - a) A nível de ambiente de rede local e/ou *internet*, as conexões que usam o SSL são as mais seguras pois evitam que programas de *sniffing* possam rastrear os dados de conexão do sistema.
4. Não foi verificada a hipótese de garantia de disponibilidade de dados e não perda dos mesmos através da replicação pois não foi possível implementar um sistema de replicação em tempo real visto que a COSYS tem comumente usado o sistema de *backup's* de toda infraestrutura em espaços de horas definidos.
5. Ao usar o PLPGSQL para criar funções e armazená-las na base de dados, torna-se o sistema mais flexível e rápido, podendo-se mais tarde migrar para outra ligação sem ser necessário reescrever a base de dados nem suas fórmulas de cálculos de impostos.

6. Por fim, a elaboração de relatórios, processo que levava minutos e até horas para concluir, agora ficou facilitada, sendo apenas necessário alguns segundos para o sistema extrair a informação correcta e actualizada sem esforço algum.

## 5. RECOMENDAÇÕES

O produto de *software* denominado “coFactura”, fruto desta pesquisa, é um projecto tecnológico super ambicioso que pode evoluir para um ERP. Para tal, as recomendações a ter em conta são:

- ✓ Formação de uma equipe que velará pela reengenharia de processos do coFactura e/ou desenvolvimento de novas funcionalidades;
- ✓ A implementação de um sistema auxiliar de replicação da base de dados para evitar a perda de dados e/ou indisponibilidade da aplicação por diversos motivos;
- ✓ A implementação de um serviço de email’s para o envio automático das facturas correspondentes aos clientes;
- ✓ Manutenção da infraestrutura de *hardware* e ambiente de software para o que o coFactura mantenha um óptimo desempenho.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

APPOLINÁRIO, F. *Dicionário de Metodologia Científica: Um Guia para a Produção do Conhecimento Científico*. São Paulo: Atlas, 2004.

BARROS, A. J. S.; LEHFELD, N. A. S. *Fundamentos de Metodologia: Um Guia para a Iniciação Científica*. 2ª Edição. São Paulo, Makron Books, 2000.

KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering: Processes and Techniques (Worldwide Series in Computer Science)*. Wiley, 1998.

KROENKE, David. *Management Information Systems*. São Paulo, McGraw-Hill, 1992.

KRUCHTEN, Philippe. *Introdução ao RUP Rational Unified Process*. Ciência Moderna, 2003.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. *Fundamentos de Metodologia Científica*. 5ª Edição. São Paulo, Editora Atlas, S.A, 2003.

MOÇAMBIQUE. Assembleia da República. *Lei n.º 32/2007 de 31 de Dezembro*. Boletim da República, Maputo, n.º 52, 2007, I Série.

MOÇAMBIQUE. Assembleia da República. *Lei n.º 03/2017 de 9 de Janeiro*. Boletim da República, Maputo, n.º 5, 2017, I Série.

MORESI, Eduardo. *Metodologia da Pesquisa. Dissertação*. Brasília, Universidade Católica de Brasília, 2003.

OLIVEIRA, M. M. *Como Fazer Pesquisa Qualitativa*. Petrópolis, Editora Vozes, 2007.

POBLACIÓN, Dinah Aguiar; WITTER, Geraldina Porto; SILVA, José Fernando Modesto da (Org.). *Comunicação e Produção Científica: Contexto, Indicadores, Avaliação*. São Paulo, Angellara, 2006.

PRESSMAN, Roger S. *Engenharia de Software*. 6ª Edição, São Paulo: McGrawHill, 2006.

SILVA, Alberto Manuel Rodrigues da; VIDEIRA, Carlos Alberto Escaleira. *UML, Metodologias e Ferramentas CASE*. 1ª Edição. Lisboa, Edições Centro Atlântico, 2001.

STEVENSON, Dennis; *Independent ICT in Schools Commission. Information and Communications Technology in UK Schools: An Independent Inquiry*. Great Britain, Independent ICT in Schools Commission, 1997.

### Fontes da Internet

*Banco de dados*. **Wikipedia**, 2003. Disponível em <[https://pt.wikipedia.org/wiki/Banco\\_de\\_dados](https://pt.wikipedia.org/wiki/Banco_de_dados)>. Acesso em: 15 de Dezembro de 2020.

*Ciclo de Vida Iterativo e Incremental*. **Wiki IFSC do Campus São José**, 2006. Disponível em: <[http://wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo\\_de\\_Vida\\_Iterativo\\_e\\_Incremental](http://wiki.sj.ifsc.edu.br/wiki/index.php/Ciclo_de_Vida_Iterativo_e_Incremental)>. Acesso em: 13 de Outubro de 2020.

*Desenvolvimento iterativo e incremental*. **Wikipedia**, 2010. Disponível em <[http://pt.wikipedia.org/wiki/Desenvolvimento\\_iterativo\\_e\\_incremental#ITERATIVO\\_x\\_INTE\\_RATIVO](http://pt.wikipedia.org/wiki/Desenvolvimento_iterativo_e_incremental#ITERATIVO_x_INTE_RATIVO)>. Acesso em: 20 de Outubro de 2014.

FABRI, José Augusto. . *Os modelos evolutivos de processo de software – A aplicabilidade do modelo incremental*. **Engenharia de Software**, 2013. Disponível em <<http://engenhariasoftware.wordpress.com/2013/01/26/os-modelos-evolutivos-de-processo-de-software-a-aplicabilidade-do-modelo-incremental/>>. Acesso em: 13 de Outubro de 2020.

*HTTP and HTTPS explained*. **Study-CCNA**, s.d. Disponível em <<https://study-ccna.com/http-https/>>. Acesso em: 21 de Outubro de 2021.

*Modelo Iterativo e Incremental*. **Revista Voat**, 2013. Disponível em: <<http://voat.com.br/rdal/?tag=incremental>>. Acesso em: 20 de Outubro de 2014.

*Modelos de Questionários Grátis.* **Survio**, 2012. Disponível em <<http://www.survio.com/pt/modelos-de-questionarios>>. Acesso em: 20 de Outubro de 2015.

*O que é uma Factura?* **Vendus**, 2019. Disponível em <<https://www.vendus.com/mz/blog/factura/>>. Acesso em: 1 de Outubro de 2020.

PACIEVITCH, Thais. *Tecnologia da Informação e Comunicação.* **InfoEscola**, 2018. Disponível em <<https://www.infoescola.com/informatica/tecnologia-da-informacao-e-comunicacao>>. Acesso em: 13 de Outubro de 2020.

*PHP - HYPERTEXT PREPROCESSOR.* **Monografias Brasil Escola**, 2006. Disponível em <<https://monografias.brasilecola.uol.com.br/computacao/php--hypertext-preprocessor.htm>>. Acesso em: 15 de Dezembro de 2020.

## **7. APÊNDICES**

## Apêndice 1: Regras de Validação de Dados no Sistema

Para a garantia do correcto uso do sistema e confiabilidade nos dados por este armazenado, deve-se garantir a observância do seguinte:

- ✓ Todos os campos obrigatórios nos formulários são assinalados com o símbolo (\*);
- ✓ O NUIT, atributo obrigatório, deve conter 9 dígitos numéricos;
- ✓ Todas as datas devem ser apresentadas no formato “DD-MM-YYYY”;
- ✓ Não usar a tecla “Enter” do teclado, sendo, portanto, que havendo necessidade de gravar alguma informação, apontar com *mouse* no botão correspondente existente no formulário;
- ✓ Em caso de falha e/ou necessidade de actualização de dados de cliente, solicitar ao gestor responsável.

## Apêndice 2: Cálculo de IVA usando *Stored Procedure*

### Definição

The screenshot shows the PostgreSQL interface for defining a function. The function name is 'calcula\_iva', it takes one argument 'montante' of type 'numeric', and returns a 'numeric' value. The programming language is 'plpgsql'. The function definition is as follows:

```
1 declare
2   valor alias for $1;
3   iva numeric (15,2);
4
5 begin
6   if coalesce(valor) > 0 then
7     iva = valor * 0.17;
8   end if;
9
10  return iva;
11 end;
```

Additional details shown include: Execution Cost: 100, Result Rows: 0, and Properties: VOLATILE, CALLED ON NULL INPUT, SECURITY INVOKER, Owner: cosys.

### Teste e *Output*

The screenshot shows a query being executed in PostgreSQL: `select calcula_iva(100) from usuario;`. The output shows a single row with the value 17.00.

#	calcula_iva
1	17.00

### Apêndice 3: Formulário que Lista os Recibos Emitidos

HOME > Administração > Lista de Recibos Emitidos ADMINISTRADOR DO PORTAL

+ Novo

Pesquisa

ID	Sigla do Cliente	Nome do Cliente	Montante	Data Emissão		
Nenhum resultado encontrado						

Sem info

Início Anterior Próximo Fim

### Apêndice 4: Formulário de Emissão de Recibo

HOME > Administração > Emissão de Recibo ADMINISTRADOR DO PORTAL

Ano Efectivo: \*

Cliente: \*

Montante: \*

Forma de Pagto: \*     Conta Bancária: \*     Nro Comprovativo: \*     Data de Pagto: \*

Emolumento: \*

Ordem	Nome do Emolumento	Valor	Eliminar
1	Custo Anual de Uso do Sistema eDondzo	250.000,00	<input type="checkbox"/>
<b>Total.....</b>		<b>250.000,00</b>	

### Apêndice 5: Formulário de Extração de Relatório de Facturas Emitidas

HOME > Relatórios > Lista de Facturas Emitidas ADMINISTRADOR DO PORTAL

Cliente: \*

Data Inicial: \*     Data Final: \*     Status: \*

- Todos --
- Pendentes
- Pagas Parcialmente
- Pagas

## Apêndice 6: Output de Relatório de Facturas Emitidas

Consultoria e Sistemas de Inform:

Facturas Emitidas de 01-06-2021 À 30-06-2021

Data	Tipo de Mov.	Sigla	Nome do Cliente	Importância
10-06-2021	Facturação	ISCISA	Instituto Superior de Ciências de Saúde	90,455.00
15-06-2021	Facturação	ISDB	Instituto Superior Dom Bosco	11,400.00


( ) Total no Período..... 101,855.00

## Apêndice 7: Formulário de Extracção de Movimentos por Conta Bancária

HOME > Relatórios > Extracto por Conta Bancária ADMINISTRADOR DO PORTAL

Conta Bancária: \*

Data Inicial: \*  Data Final: \*



## Apêndice 8: Output de Extracção de Movimentos por Conta Bancária

COSYS – Consultoria e Sistemas de Informação, Lda

SECTOR: ADMINISTRAÇÃO

LIVRO DE CONTROLE DA CONTA BANCÁRIA Nº 1111101200 – BCI

PARA O PERÍODO DE 01-01-2021 À 10-03-2021

VALORES EXPRESSOS EM MZN

Data	Tipo de Mov.	Nº Comprovativo	Sigla	Nome do Cliente	Importância	Saldo
01-01-2021					500,000.00	500,000.00
26-01-2021	Entrada	'20013011180007	ISCISA	Instituto Superior de Ciências de Saúde	70,000.00	570,000.00
02-02-2021	Entrada	'20013012030786	USTM	Universidade São Tomé de Moçambique	180,400.00	750,400.00
04-02-2021	Entrada	'20013015175520	ISCISA	Instituto Superior de Ciências de Saúde	95,000.00	845,400.00
01-03-2021	Entrada	'20013014101871	ISPG	Instituto Superior Politécnico de Gaza	10,000.00	855,400.00
06-03-2021	Entrada	'20030610404886	ISDB	Instituto Superior Dom Bosco	11,400.00	866,800.00

( )


Saldo Inicial..... 500,000.00  
 Entradas..... 366,800.00  
 Saldo Actual.... 866,800.00

## Apêndice 9: Formulário de Extração de Situação Financeira do Cliente

HOME > Relatórios > Extracto por Cliente ADMINISTRADOR DO PORTAL

Cliente: \* -- Todos --

Data Inicial: \* 01-01-2021      Data Final: \* 02-12-2021



## Apêndice 10: Output de Extração de Situação Financeira do Cliente

Ord	Sigla	Nr Factura	Débito	Crédito	Saldo
1	ISCAM	00138	250,000.00	100,000.00	150,000.00
2	ISCISA	00097	120,000.00		120,000.00
3	ISCISA	00098	50,000.00	50,000.00	-
4	ISGN	00005	350,000.00	300,000.00	50,000.00
5	ISPTEC	00099	150,000.00	125,000.00	25,000.00
6	USTM	00015	90,675.00	30,000.00	60,675.00
<b>Total....</b>			<b>1,010,675.00</b>	<b>605,000.00</b>	<b>405,675.00</b>

## Apêndice 11: Factura



Boane, Matola-Rio, Comunidade de Djuba  
 Cel: +258 82 3030993 Fax: 21019812  
 E-mail: info@cosys.co.mz  
**NUIT: 400403333**  
 Maputo - Moçambique

Exmo(s). Sr.(s) **USTM**

Morada:

.....

Tel: ..... Fax: .....

**NUIT:**

**FACTURA**  
**Nº 00015**

MATOLA

15 de Junho de 2020

QTD.	DESCRIÇÃO	P. UNITÁRIO	VALOR TOTAL
1	Transporte Local	10,000.00	10,000.00
1	Ajuda de Custos	15,000.00	15,000.00
1	Alojamento	18,000.00	18,000.00
1	Comunicação	34,500.00	34,500.00
<b>Sub - Total</b>			77,500.00 Mt
MOTIVO justificativo da não aplicação do Imposto			13,175.00 Mt
17 % IVA			
<b>TOTAL</b>			90,675.00 Mt

São: **Noventa mil, seiscentos e setenta e cinco meticais**

Detalhes para o Pagamento:

BANCO: BCI - Banco Comercial e de Investimentos  
 NIB: 0008 0000 1111 101200 80  
 Conta: 1111101200

Administrador do Sistema

Impressão dia 21-11-2021

coFactura - Processado por computador

<https://cosys.co.mz>

<https://portal.cosys.co.mz>

## Apêndice 12: Recibo



**COSYS ? Consultoria e Sistemas de Informação, Lda**

Bairro Mozal, Djuba, Quarteirão 1, Talhão 54, Boane - Maputo

Contacto: 82 30 30 993 ou 84 71 31 031

**NUIT: 400 403 333**

Recibo n°: **00250/2021**

Emitido em: 10:50 10-10-2021

Recebemos de: **Universidade São Tomás de Moçambique** NUIT **700057585**

Valor	Forma Pagamento	N° do Talão	Data talão	Conta
30,000.00	Depósito Bancário	215554112541	10-10-2021	9952665710001 - BCI

Extenso: São trinta mil meticais

Ordem	Referente a:	Valor:
1	Comunicação	5,000.00
2	Transporte Local	10,000.00
3	Ajudas de Custo	15,000.00
<b>Total....</b>		<b>30,000.00</b>

Impresso no dia 28-11-2021

Administrador do Sistema

coFactura - Processado por computador

<https://cosys.co.mz>

<https://portal.cosys.co.mz>



**COSYS ? Consultoria e Sistemas de Informação, Lda**

Bairro Mozal, Djuba, Quarteirão 1, Talhão 54, Boane - Maputo

Contacto: 82 30 30 993 ou 84 71 31 031

**NUIT: 400 403 333**

Recibo n°: **00250/2021**

Emitido em: 10:50 10-10-2021

Recebemos de: **Universidade São Tomás de Moçambique** NUIT **700057585**

Valor	Forma Pagamento	N° do Talão	Data talão	Conta
30,000.00	Depósito Bancário	215554112541	10-10-2021	9952665710001 - BCI

Extenso: São trinta mil meticais

Ordem	Referente a:	Valor:
1	Comunicação	5,000.00
2	Transporte Local	10,000.00
3	Ajudas de Custo	15,000.00
<b>Total....</b>		<b>30,000.00</b>

Impresso no dia 28-11-2021

Administrador do Sistema

coFactura - Processado por computador

<https://cosys.co.mz>

<https://portal.cosys.co.mz>

## Apêndice 13: Avaliação do Software

### **Avaliação do Software coFactura – Gestão de Facturação**

O objectivo deste inquérito é avaliar o processo de instalação e a usabilidade do coFactura, sistema de gestão de facturação implementado na COSYS entre os meses de Março e Abril de 2021. Pretende, também, avaliar o grau de alcance dos objectivos preconizados aquando da proposta de desenvolvimento e implementação deste software.

1. A instalação do software foi:

- Muito fácil
- Mais ou menos fácil
- De dificuldade média
- Mais ou menos difícil
- Muito difícil

2. O processo da instalação do software foi:

- Rápido
- Normal
- Lento

3. A interface do software é amigável (software fácil de usar)?

- Sim, muito
- Mais ou menos sim
- De dificuldade média
- Mais ou menos não
- Absolutamente não

4. A documentação que acompanha o software é:

- Muito útil
- Mais ou menos útil

- Normal
  - Mais ou menos inútil
  - Absolutamente inútil
5. Em que medida é útil a assistência técnica do desenvolvedor em relação ao software?
- Muito útil
  - Mais ou menos útil
  - Normal
  - Mais ou menos inútil
  - Absolutamente inútil
6. Com que frequência “congela” ou falha o software?
- Muito frequentemente
  - Frequentemente
  - Às vezes
  - Quase nunca
  - Nunca
7. Como está satisfeito/a com o funcionamento do software?
- Muito satisfeito/a
  - Satisfeito/a
  - Médio satisfeito/a
  - Insatisfeito/a
  - Muito insatisfeito/a
8. Recomendaria este software à outras instituições?
- Definitivamente sim
  - Provavelmente sim
  - Não sei

- Provavelmente não
- Definitivamente não

9. Comentários sobre o software

---

---

---

---

---

---

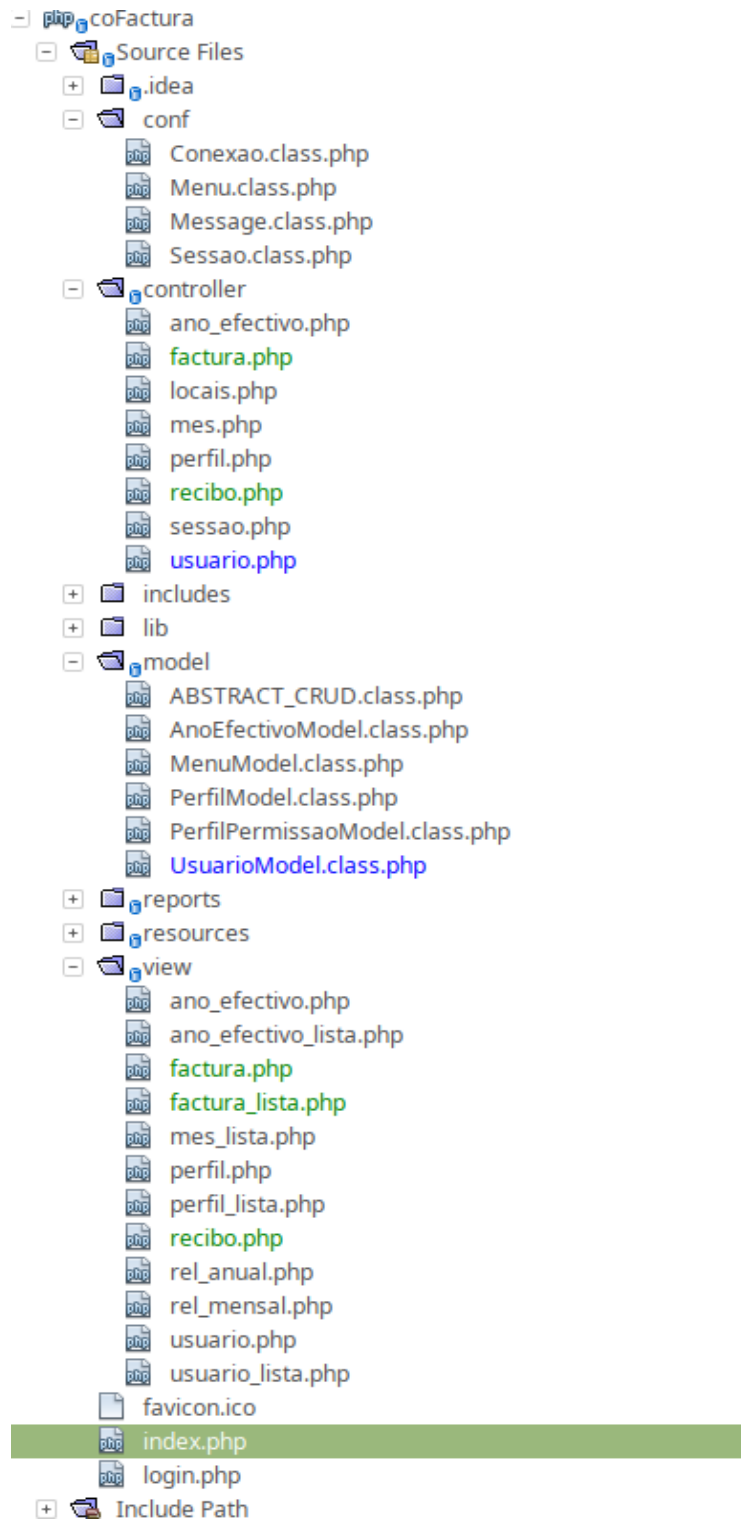
---

Representante da COSYS

---

Maputo, aos 2 de Dezembro de 2021

## Apêndice 14: Estrutura de Ficheiros do coFactura



Apêndice 15: Perfil do Administrador do coFactura



## **PERFIL DO ADMINISTRADOR DO SISTEMA coFactura**

- Importante:
  - ✓ Conhecimento do sistema financeiro moçambicano;
  - ✓ Experiência em administração de sistemas corporativos constitui vantagem;
  - ✓ Capacidade de trabalhar sob pressão;
- Recomendados:
  - ✓ Conhecimento sobre configuração, optimização e administração de servidores de base de dados;
  - ✓ Capacidade de leitura e interpretação de logs tanto do sistema operativo quanto das aplicações;
  - ✓ Fortes conhecimentos sobre firewall e proxies;
  - ✓ Fortes conhecimentos sobre redes de computadores.

### Responsabilidades:

- Garantia da disponibilidade do sistema;
- Monitorar a correcta criação de backup's diários ou sob qualquer periodo de tempo estipulado;
- Monitorar o acesso ao sistema e verificar aspectos de segurança do mesmo;
- Gerir usuários do Sistema;
- Gerir e definir Perfis de Utilizadores do Sistema;
- Dar formação e suporte aos usuários a todos os níveis e funcionalidades do sistema;
- Coordenar a (re)engenharia de requisitos do Sistema, com os gestores da COSYS;

Maputo, 30 de Agosto de 2021

\_\_\_\_\_  
(António Ricardo Guirruço)

N.º Iteração	Módulo	Funcionalidade	Data	Status
1	Administração do Sistema	Gerir Perfis dos Usuários	10-03-2021	Feito
		Gerir Usuários	10-03-2021	Feito
		Gerir Anos Efectivos	10-03-2021	Feito
		Gerir Emolumentos	10-03-2021	Feito
		Gerir Bancos	10-03-2021	Feito
		Gerir Contas Bancárias	10-03-2021	Feito
		Gerir Clientes	10-03-2021	Feito
2	Operações Correntes	Gerir Facturas	26-03-2021	Feito
		Gerir Recibos	26-03-2021	Feito
3	Relatórios	Facturas	01-04-2021	Feito
		Extracto por Cliente	01-04-2021	Feito
		Extracto por Conta Bancária	01-04-2021	Feito



## Apêndice 19: Cronograma de Entregas (Desenvolvimento vs Funcionalidades)

N.º Iteração	Módulo	Funcionalidade	Data	Status
1	Administração do Sistema	Gerir Perfis dos Usuários	10-03-2021	Feito
		Gerir Usuários	10-03-2021	Feito
		Gerir Anos Efectivos	10-03-2021	Feito
		Gerir Emolumentos	10-03-2021	Feito
		Gerir Bancos	10-03-2021	Feito
		Gerir Contas Bancárias	10-03-2021	Feito
		Gerir Clientes	10-03-2021	Feito
2	Operações Correntes	Gerir Facturas	26-03-2021	Feito
		Gerir Recibos	26-03-2021	Feito
3	Relatórios	Facturas	01-04-2021	Feito
		Extracto por Cliente	01-04-2021	Feito
		Extracto por Conta Bancária	01-04-2021	Feito

## Apêndice 20: Exemplo de Código-Fonte de uma View

```
Source History
1 <?php
2 require '../conf/Sessao.class.php';
3 require '../conf/Conexao.class.php';
4
5 $DBS = new Conexao();
6 $con = $DBS->getCon();
7 ?>
8 <form id="frmUsuario" class="login-form">
9
10 <section id="erro"></section>
11
12 <?php
13 if (isset($_POST['id']))(
14
15     $id = trim($_POST['id']);
16
17     $dd = $con->query("select * from usuario where id='".$id."'")->fetch(PDO::FETCH_ASSOC);
18
19     echo '<input type="hidden" name="id" value="'.$id.'"/>';
20     echo '<input type="hidden" name="acao" value="edita"/>';
21 }elseif(
22     echo '<input type="hidden" name="acao" value="grava"/>';
23 }
24 ?>
25
26 <section class="row-fluid form-horizontal">
27 <section class="span12 well">
28 <section class="row-fluid">
29 <section class="span12 bgcolor">
30 <section class="control-group">
31 <label class="control-label" for="login">Login: </label>
32 <section class="controls">
33 <?php
34 if (isset($_POST['id']))(
35     echo '<input type="tel" id="login" name="login" placeholder="login" maxlength="20" class="span4" value="'.$dd['id'].'." disabled="false">';
36 } else {
37     echo '<input type="tel" id="login" name="login" placeholder="login" maxlength="20" class="span4" value="" autofocus="autofocus">';
38 }
39 ?>
40 </section>
41 </section>
42
43 <?php
44 if (isset($_POST['id']))(
45 ?>
46 <section class="control-group">
47 <label class="control-label" for="nome">Senha: </label>
48 <section class="controls">
49 <input type="tel" disabled placeholder="login" maxlength="20" class="span4" value="" >
50 </section>
51 </section>
52
53 <?php
54 }elseif ?>
55
56 <section class="control-group">
57 <label class="control-label" for="senha_muda">Mudar Senha: </label>
58 <section class="controls">
59 <input type="tel" id="senha_muda" name="senha_muda" placeholder="Nova Senha" maxlength="20" class="span4" value="" >
60 </section>
61 </section>
62
63 <?php ?>
64
65 <section class="control-group">
```

## Apêndice 21: Exemplo de Código-Fonte do Controller

```

1 <?php
//Controller que fará a implementação da classe Usuário através do hint: accao
require $_SERVER['DOCUMENT_ROOT'].'/cofactura/model/UsuarioModel.class.php';

1 class Usuario extends UsuarioModel{
. }

1 if (isset($_POST['acao'])){
    $acao = $_POST['acao'];
    $usuario = new Usuario();

1     if ($acao == 'grava'){
        $id = $_POST['login'];
        $perfil = $_POST['perfil'];
        $telefone = $_POST['cell'];
        $ativo = $_POST['ativo'];
        $nome = $_POST['nome'];

        $usuario->setId($id);
        $usuario->setCell($telefone);
        $usuario->setActivo($ativo);
        $usuario->setNome($nome);
        $usuario->setPerfilId($perfil);
        echo $usuario->grava();
    }

1     if ($acao == 'elimina'){
        $id = $_POST['id'];
        $usuario->setId($id);
        echo $usuario->elimina();
    }
. }
2 >

```

## Apêndice 22: Exemplo de Código-Fonte de Model

```

<?php
require 'ABSTRACT_CRUD.class.php';
/**
 *
 * @version 1.107
 * @package entity
 */
class UsuarioModel extends ABSTRACT_CRUD {
    private $id;
    private $nome;
    private $senha;
    private $email;
    private $perfilId;
    private $tipo;
    private $traceId;
    private $cell;
    private $dataReg;
    private $ativo;
    public $erro;
    private $logado;

    public function getId() {
        return $this->id;
    }

    public function getNome() {
        return $this->nome;
    }

    public function getSenha() {
        return $this->senha;
    }

    public function getEmail() {
        return $this->email;
    }

    public function getPerfilId() {
        return $this->perfilId;
    }

    public function getTipo() {
        return $this->tipo;
    }

    public function getTraceId() {
        return $this->traceId;
    }

    public function getCell() {
        return $this->cell;
    }

    public function getDataReg() {
        return $this->dataReg;
    }

    public function grava() {
        $erro = '';

        $erro = '';
        if (trim($this->id)){
            $erro .= '<li>Login não informado!</li>';
        }

        if (trim($this->nome)){
            $erro .= '<li>Nome não informado!</li>';
        }

        if (trim($this->perfilId)){
            $erro .= '<li>Perfil não informado!</li>';
        }

        if (trim($erro)){
            $ex = new Message($erro, 'gravar');
            return $ex->raiseError();
        }else{
            $DBS = new Conexao();
            $con = $DBS->getCon();
            $this->nome = strtoupper($this->nome);
            $id = $con->quote($this->id);

            if ($this->ativo == '1'){
                $this->ativo = 'TRUE';
            } else {
                $this->ativo = 'FALSE';
            }

            $si = $con->query("select count(*) from usuario where id = ".$id."")->fetchColumn();


            if ($si >=0){
                $erro = "<li>Login já está em uso no sistema</li>";
                $ex = new Message($erro, 'gravar');
                return $ex->raiseError();
            }else{
                try{
                    $stm = $con->prepare("insert into usuario (id, nome, perfil_id, cell, ativo, trace_id)
                    . "values (:id, :nome, :perfil_id, :cell, :ativo, :trace_id)");
                    $stm->bindParam(':id', $this->id);
                    $stm->bindParam(':nome', $this->nome);
                    $stm->bindParam(':perfil_id', $this->perfilId);
                    $stm->bindParam(':cell', $this->cell);
                    $stm->bindParam(':ativo', $this->ativo);
                    $this->traceId = $_SESSION['staff_id'];
                    $stm->bindParam(':Trace_id', $this->traceId);
                    $stm->execute();
                    $msg = new Message('Usuário', 'gravado');
                    return $msg->raiseSuccess();
                } catch (Exception $ex) {
                    $err = new Message($ex, 'gravar');
                    return $err->raiseError();
                }
            }
        }

        public function lista() {

```

**8. ANEXOS**

Anexo 1: Factura

 <b>OSYS</b> Consultoria & Sistemas Boane, Matola-Rio, Comunidade de Djuba Cel: +258 82 3030993 Fax: 21019812 E-mail: info@cosys.co.mz NUIT: 400403333 Maputo - Moçambique	Exmo(s). Sr.(s) <b>USTM</b> Morada: ..... Telf: ..... Fax: ..... <b>NUIT:</b>	<b>FACTURA</b> <b>Nº 00015</b>
	MATOLA	

15 de Junho de 2020

QUANT.	DESCRIÇÃO	P. UNITÁRIO	VALOR TOTAL
1	Transporte Local	10,000.00	10,000.00
1	Ajuda de Custos	15,000.00	15,000.00
1	Alojamento	18,000.00	18,000.00
1	Passagens Áreas	0.00	0.00
1	Comunicação	34,500.00	34,500.00
São: <b>Noventa mil, seiscientos e setenta e cinco meticais</b>			

Processado por computador	<b>Sub - Total</b>	77,500.00 Mt
MOTIVO justificativo da não aplicação do Imposto ..... .....	<b>17 % IVA</b>	13,175.00 Mt
	<b>TOTAL</b>	90,675.00 Mt

<b>Contas Bancárias</b>	
BCI:	
NIB: <b>0008 0000 1111101200 80</b>	
Conta: <b>1111101200</b>	

Anexo 2: Alvará da COSYS



REPÚBLICA DE MOÇAMBIQUE

GOVERNO DO DISTRITO DE BOANE

SERVIÇO DISTRITAL DE ACTIVIDADES ECONOMICAS DE BOANE

**EXERCÍCIO DE ACTIVIDADE COMERCIAL**

ALVARÁ Nº: **386/10/02/PS/2018**

DECRETO Nº 34/2013, de 2 de Agosto

Titular: **COSYS CONSULTORIA E SISTEMA DE INFORMACAO SOCIEDADE UNIPessoal LDA**

NUIT: **400403333**

Actividade Principal: **62010 - ACTIVIDADES DE PROGRAMAÇÃO INFORMÁTICA**

Endereço Estabelecimento(s) **BAIRRO DJUBA, BOANE, MAPUTO**

Validade: **POR TEMPO INDETERMINADO**

Para constar, se lavrou o presente Alvará que é por mim assinado e segue devidamente autenticado,

**Maputo, 01 de Outubro de 2018**

A handwritten signature in black ink, appearing to read 'Teresa Helena Boaventura Mauaie', written over a horizontal line.

**Teresa Helena Boaventura Mauaie**  
**Administradora**

004772018027880016829538  
Conferir no Serviço de Contra Prova no Portal do Cidadão