

**SISTEMA DE GESTÃO DE CONTAS BANCÁRIAS PARA
EMPRESA ORGANIZAÇÕES MAMADE HANIF**

Trabalho de investigação a ser submetido de acordo com os requisitos e
exigências para obtenção do grau de licenciatura

de

IZDIN ABDUL GAFAR

Estudante Nr. 194280

DO CURSO DE INFORMÁTICA DE GESTÃO

da

FACULDADE DE CIÊNCIAS E TECNOLOGIAS

da

UNIVERSIDADE POLITÉCNICA



Supervisor: Lic. Stélio Rosa Loforte

Maputo, 27 de Março de 2014

DECLARAÇÃO DE HONRA

Eu, Izdin Abdul Gafar, declaro que este trabalho de fim de curso foi exclusivamente realizado por mim. O mesmo é agora submetido de acordo com todos os requisitos e exigências para a obtenção do grau de Licenciatura em Informática de Gestão, na Universidade Politécnica em Maputo.

Assinatura _____

Data ___/___/_____

PARECER DO TUTOR

O trabalho de investigação do estudante Izdin Abdul Gafar cujo tema é **Sistema De Gestão De Contas Bancárias Para Empresa OMH**, foi por mim acompanhado durante a sua elaboração.

O estudante mostrou vontade e dedicação tendo apresentado o trabalho várias vezes correspondendo a cada etapa da sua elaboração.

O mesmo analisou o funcionamento do sistema bancário da empresa acima citada tendo indicado os constrangimentos e soluções para os mesmos e apresentado um protótipo do sistema.

O tema é actual, pertinente e vital importância para organizações que trabalham com Bancos.

O trabalho elaborado apresenta os requisitos necessários para ser submetido para a obtenção do grau de licenciatura.

Maputo, aos 27 de Março de 2014

O Tutor

(Lic. Stélio Rosa Loforte)

DEDICATÓRIA

Dedico este trabalho aos meus pais Abdul Gafar e Paula Varela, por terem me ensinado a aprender, lutar e nunca deixar de acreditar nos meus sonhos. Onde quer que eles estejam, tenho a certeza de que estarão acompanhando esta realização.

AGRADECIMENTOS

Agradeço a Deus por todos os momentos de força e esperança. À minha família, em especial à minha mãe Paula, por todo amor, carinho, apoio, incentivo incondicional e compreensão pela distância necessária para a realização da minha licenciatura, distancia essa que me manteve afastado dos seus últimos momentos de vida. Ao meu pai obrigado por tudo, inclusive pelo apoio financeiro! À minha irmã, Rosymin pelo amor e carinho sempre presentes mesmo com a distância.

Ao meu tutor, Prof. Dr. Stélio Rosa Loforte pela orientação, confiança e incentivo durante todas as fases deste trabalho. Sua experiência e sugestões foram essenciais para a realização deste trabalho.

Aos meus amigos e colegas de jornadas Ali, Yudi, Marco e Taibo pela amizade, parceria durante a Licenciatura e também por estarem sempre disponíveis em quaisquer momentos. A todos os funcionários, professores e estudantes que participaram e colaboraram de alguma maneira na realização deste trabalho.

E por ultimo, mas definitivamente não menos importante, o meu grande obrigado a empresa Organizações Mamade Hanif e os seus funcionários pela confiança em mim depositada.

ÍNDICE

| | |
|--|-------------|
| Declaração de Honra | ii |
| Parecer do Tutor | iii |
| Dedicatória | iv |
| Agradecimentos | v |
| Índice | vi |
| Lista de Figuras | viii |
| Lista de Tabelas | ix |
| Lista de Siglas | x |
| Abstracto | xi |
| CAPITULO 1 - INTRODUÇÃO | 1 |
| 1.1. A pergunta a investigar e as hipóteses a considerar | |
| 1.2. A metodologia de investigação | |
| 1.3. Os objectivos da investigação | |
| 1.4. A importância do tema proposto para investigação | |
| 1.5. Conteúdo do trabalho | |
| CAPITULO 2 - LEITURA BIBLIOGRÁFICA | 5 |
| 2.1. Informação e conhecimento | |
| 2.2. Tecnologias de Informação | |
| 2.3. Sistemas de Informação | |
| 2.4. Análise e desenvolvimento de softwares | |
| 2.5. Metodologias de desenvolvimento de Sistemas | |
| CAPITULO 3 - A EMPRESA | 16 |
| 3.1. Organizações Mamade Hanif | |
| 3.2. A empresa antes do Sistema | |
| 3.3. A empresa depois do Sistema | |
| CAPITULO 4 - REQUISITOS E MODELAGEM | 18 |
| 4.1. Requisitos do Sistema | |
| 4.2. Diagramas de Modelagem | |
| CAPITULO 5 - O SISTEMA | 39 |
| 5.1. Acesso ao bancOMH | |
| 5.2. Módulos de gestão | |

| | |
|---|-----------|
| 5.3. Movimentos | |
| 5.4. Visualizar | |
| 5.5. Definições | |
| 5.6. Instalação do Sistema | |
| CAPITULO 6 - CONCLUSÕES E RECOMENDAÇÕES | 60 |
| 6.1. Recomendações | |
| 6.2. Trabalhos futuros | |
| ANEXO A – RELATÓRIO DE CHEQUES PASSADOS | 62 |
| ANEXO B – RELATÓRIO DE DEPÓSITOS EM CHEQUE | 63 |
| ANEXO C – RELATÓRIO DE SALDOS | 64 |
| ANEXO D – RELATÓRIO DE EXTRACTOS | 65 |
| ANEXO E – PLANO DE ENTREVISTA | 66 |
| BIBLIOGRAFIA | 67 |

LISTA DE FIGURAS

| | |
|---|----|
| FIGURA 1 - DIAGRAMA DE CASOS DE USO | 20 |
| FIGURA 2 - PACOTE DE GESTÃO DE CIDADES | 21 |
| FIGURA 3- PACOTE DE GESTÃO DE USUÁRIOS | 21 |
| FIGURA 4 - PACOTE DE GESTÃO DE CONTAS..... | 22 |
| FIGURA 5 - PACOTE DE GESTÃO DE MOVIMENTOS..... | 22 |
| FIGURA 6 – DIAGRAMA DE CLASSES..... | 36 |
| FIGURA 9 - TELA DE CARREGAMENTO..... | 40 |
| FIGURA 10– TELA DE LOGIN..... | 41 |
| FIGURA 11– TELA PRINCIPAL DO BANCO MH | 42 |
| FIGURA 12– TELA DAS CIDADES..... | 42 |
| FIGURA 13– CONFIRMAÇÃO DE ELIMINAÇÃO DE CIDADE..... | 44 |
| FIGURA 14– MENSAGEM DE ELIMINAÇÃO SEM EFEITO..... | 44 |
| FIGURA 15– TELA DOS USUÁRIOS | 46 |
| FIGURA 16– TELA DE ADIÇÃO DE CIDADES | 46 |
| FIGURA 17– TELA DAS MOEDAS | 47 |
| FIGURA 18– TELA DOS BANCOS | 48 |
| FIGURA 19– TELA DAS GESTÃO DAS CONTAS..... | 49 |
| FIGURA 20– TELA DOS CHEQUES PASSADOS | 51 |
| FIGURA 21– TELA DOS DEPÓSITOS | 52 |
| FIGURA 22– TELA DE OUTROS MOVIMENTOS | 53 |
| FIGURA 23– TELA DAS ACTUALIZAÇÕES DE MOVIMENTOS | 54 |
| FIGURA 24– TELA DE EDIÇÃO DE UM MOVIMENTO | 55 |
| FIGURA 25– PESQUISA DE CHEQUES PASSADOS | 56 |
| FIGURA 26– PESQUISA DE DEPÓSITOS EM CHEQUE..... | 57 |
| FIGURA 27– PESQUISA DE EXTRACTO DE CONTA | 57 |
| FIGURA 28– PESQUISA DE SALDOS PARA CIDADE SELECIONADA | 58 |
| FIGURA 29– TELA “SOBRE” | 58 |

LISTA DE TABELAS

| | |
|---|----|
| TABELA 1 - DESCRIÇÃO DO CASO DE USO ADICIONAR CIDADE..... | 24 |
| TABELA 2 - DESCRIÇÃO DO CASO DE USO EDITAR CIDADE..... | 25 |
| TABELA 3 - DESCRIÇÃO DO CASO DE USO APAGAR CIDADE | 26 |
| TABELA 4 - DESCRIÇÃO DO CASO DE USO ADICIONAR USUÁRIO | 27 |
| TABELA 5 - DESCRIÇÃO DO CASO DE USO EDITAR USUÁRIO..... | 28 |
| TABELA 6 - DESCRIÇÃO DO CASO DE USO APAGAR USUÁRIO | 29 |
| TABELA 7 - DESCRIÇÃO DO CASO DE USO ADICIONAR CONTA..... | 30 |
| TABELA 8 - DESCRIÇÃO DO CASO DE USO EDITAR CONTA..... | 31 |
| TABELA 9 - DESCRIÇÃO DO CASO DE USO APAGAR CONTA | 32 |
| TABELA 10 - DESCRIÇÃO DO CASO DE USO ADICIONAR MOVIMENTO..... | 33 |
| TABELA 11 - DESCRIÇÃO DO CASO DE USO EDITAR MOVIMENTO | 34 |
| TABELA 12 - DESCRIÇÃO DO CASO DE USO APAGAR MOVIMENTO | 35 |

LISTA DE SIGLAS

| | |
|--------------|--|
| CASE: | Computer-Aided Software Engineering |
| ERP: | Enterprise Resource Planning |
| ISO: | International Organization for Standardization |
| OMG: | Object Management Group |
| OMH: | Organizações Mamade Hanif |
| OMT: | Object Modeling Technique |
| SI: | Sistema de Informação |
| TI: | Tecnologias de Informação |
| UML: | Unified Modeling Language |
| VB6: | Visual Basic 6 |

ABSTRACTO

| | |
|--------------------------------|--|
| Autor: | Izdin Abdul Gafar |
| Grau Académico: | Licenciatura em Informática de Gestão |
| Título: | Sistema de Gestão de Contas Bancárias para empresa Organizações Mamade Hanif |
| Universidade: | A Politécnica |
| Faculdade: | Gestão, Ciências e Tecnologias |
| Supervisor da Proposta: | Dr. Stélio Rosa Loforte |
| Data: | 27 de Março de 2014 |
| Palavras chave: | Informatização, ERP, sistema bancário, gestão de contas, Sistema tipo <i>software</i> , uml, orientação a objectos. |

Nos dias actuais e com a evolução galopante da tecnologia e comunicação, a informatização chega às mais variadas áreas da nossa economia. Esse facto fez crescer muito a competitividade entre as empresas e forçou o surgimento de diferenciais na hora de efectuar a gestão das informações disponíveis. A empresa Organizações Mamade Hanif vê-se em desvantagem por não possuir um sistema consistente e de fácil acesso que possa auxiliar no processo de gestão das suas contas bancárias. Com este trabalho teórico-prático, pretende-se desenvolver uma aplicação multi-usuário capaz de fazer a gestão das contas bancárias da empresa OMH em número ilimitado em termos dos seus movimentos e da conferência dos mesmos, numa solução de utilização simples e prática, proporcionando suporte à tomada de decisão e, com interfaces simples e intuitivas, facilitar a vida dos usuários do sistema que, em geral, possuem pouca habilidade com sistemas de informação. Para que se alcance o sucesso desejado, será usada a metodologia de orientação a objectos com modelagem UML.

CAPITULO 1 - INTRODUÇÃO

Actualmente, com a economia e o mundo globalizado, as empresas buscam cada vez mais obter vantagem competitiva. A Tecnologia da Informação (TI), através dos Sistemas de Informação (SI), pode ser usada como uma poderosa ferramenta de diferencial competitivo no mercado actual. Os recursos da TI fornecem suporte à tomada de decisão, controle de produção e qualidade, além de vários outros níveis de gestão, oferecendo mais facilidade de administração e gestão interna das organizações.

Com o surgimento da automação industrial e do fluxo de informações internas, a TI está cada vez mais presente no ambiente corporativo, sendo essencial para qualquer actividade de gestão do mesmo. Assim, ao se obter melhor gestão da informação interna da empresa, esse fluxo harmónico se reflectirá também no ambiente externo, no nicho de mercado em que a organização se encontra e no mercado de trabalho global.

A existência de um sistema de gestão tipo ERP (*Enterprise Resource Planning*) faz com que informações, que antes estavam guardadas e de difícil acesso, estejam mais facilmente disponíveis e se tornando peça diferencial no processo de tomada de decisões e gestão do negócio. A Organizações Mamade Hanif (OMH), empresa nacional que actua no ramo de importações e exportações, sentiu-se em desvantagem por não possuir ferramentas deste tipo e vivia um cenário não muito agradável devido ao tempo despendido para a tomada de decisões e a efectividade destas.

A empresa era guiada por um sistema manual baseado em apontamentos diários que no final do dia apresentava não só inconsistências mas proporcionava também dificuldades no processo de busca de informações e criação de relatórios. Como consequência, planos de

contingência teriam de ser postos em prática, e algumas operações teriam de ser canceladas. O Problema? *A qualidade das informações disponíveis.*

1.1.A PERGUNTA A INVESTIGAR E AS HIPÓTESES A CONSIDERAR

Pergunta: *Estaria o sucesso das decisões bancárias relacionado com a qualidade das informações disponíveis?*

1.1.1. As Hipóteses

H0: O sucesso das decisões bancárias não está relacionado com a qualidade das informações disponíveis.

H1: O sucesso das decisões bancárias está relacionado com a qualidade das informações disponíveis.

1.1.2. Perguntas Investigativas de Apoio à Investigação

- Como melhorar a qualidade das informações?
- Como usar as informações disponíveis para criar diferenciais?

1.2.A METODOLOGIA DE INVESTIGAÇÃO

A investigação será conduzida por um paradigma crítico. Será usada a análise orientada a objectos a fim de obter uma visão ampla das características do sistema. Após a análise teórica dessas características, o desenho e desenvolvimento do sistema será feito usando modelagem UML (Unified Modeling Language) à qual serão adicionados o Diagrama Casos de Uso, Diagrama de Classes, Diagrama de Sequência e Diagrama Entidade-Relacionamento. A metodologia empregada, porém, não se resume somente ao desenvolvimento do sistema, mas também às entrevistas abertas feitas aos funcionários da empresa OMH e aos testes e correcções efectuadas após o desenvolvimento do projecto.

Uma vez que serão os funcionários da empresa OMH os usuários finais do sistema, é importante que estes estejam envolvidos nos processos de levantamento de dados e testagem do sistema.

Durante a fase de desenvolvimento do sistema foram usadas ferramentas *open source* na sua maioria. O sistema de gestão de base de dados escolhido é o MySQL. A linguagem de Programação usada, o Visual Basic 6 (VB6), este acompanha custos de aquisição e licenciamento. Durante este processo, todos os recursos de *hardware* usados foram disponibilizados pelo estudante.

1.3.OS OBJECTIVOS DA INVESTIGAÇÃO

1.3.1. Os objectivos gerais

O trabalho a ser desenvolvido tem como propósito principal a criação de um sistema informatizado de fácil acesso com alta persistência e consistência dos dados que ofereça auxílio na gestão das contas bancárias da empresa OMH e que seja capaz de produzir relatórios sempre que estes se façam necessários.

1.3.2. Os objectivos específicos

Com a finalidade de garantir uma maior compatibilidade com a realidade do negócio e facilitar a vida dos usuários, este sistema deveria ser capaz de:

- Guardar dados de contas Bancárias
- Guardar informações relacionadas com movimentos das contas
- Garantir aumento da rapidez e acesso aos dados
- Disponibilizar relatórios de saldos e extractos das contas
- Proporcionar segurança das informações disponíveis
- Oferecer melhorias no fluxo das actividades dentro da empresa

1.4.A IMPORTÂNCIA DO TEMA PROPOSTO PARA INVESTIGAÇÃO

O sistema a ser desenvolvido é de extrema importância, não só para a empresa OMH, mas também para outras empresas que ainda não tenham um sistema deste tipo porque o mesmo irá influenciar positivamente no rendimento dos funcionários, na disponibilidade e segurança de informação, criação de diferenciais e transparência do negócio.

1.5.CONTEÚDO DO TRABALHO

O presente trabalho está disposto em seis capítulos sendo:

- **Capítulo 1 – Introdução:** Aborda aspectos introdutórios sobre o trabalho, metodologia de investigação e o método usado para a colecta e análise dos dados, a justificativa do tema bem como os objectivos gerais e específicos da investigação.
- **Capítulo 2 – Leitura Bibliográfica:** Neste capítulo são abordados conceitos como Tecnologias de informação, sistemas de informação, Metodologias de desenvolvimento e Modelagem de sistemas.
- **Capítulo 3 – A Empresa:** Neste capítulo é apresentada a empresa Organizações Mamade Hanif (OMH), o sistema actual de gestão de suas contas bancárias, suas deficiências e as melhorias que se esperam com a implantação do sistema a ser desenvolvido.
- **Capítulo 4 – Requisitos e Modelagem:** Este capítulo apresenta a modelagem do sistema feita através da Análise Orientada a Objectos. São apresentados diagramas casos de uso, diagrama de classes, diagrama de sequências e diagrama de Entidade-Relacionamento. São também apresentados os requisitos funcionais e não funcionais do sistema.
- **Capítulo 5 – O Sistema:** Neste capítulo é apresentado o protótipo do sistema em questão, ao qual chamamos de *bancOMH* e suas funcionalidades. São também abordados os requisitos para a instalação do sistema.
- **Capítulo 6 – Conclusões e Recomendações:** Neste capítulo são abordadas as conclusões alcançadas, as mudanças que o sistema ofereceu para a empresa OMH, recomendações sobre o tema em estudo e proposta de trabalhos futuros.

CAPITULO 2 - LEITURA BIBLIOGRÁFICA

2.1.INFORMAÇÃO E CONHECIMENTO

Após a globalização e o surgimento da informática, o cenário económico mundial tornou-se mais competitivo. Passou-se de uma economia industrial para uma economia de informação, de modo que as empresas que não se informatizam perdem lugar no mercado muito rapidamente.

Segundo McGee e Prusak (1994:244), em uma economia de informação, a concorrência entre as organizações baseia-se em sua capacidade de adquirir, tratar, interpretar e utilizar a informação de forma eficaz. Aquela empresa que gerir e utilizar correctamente a sua informação interna alcançará diferencial competitivo no mercado.

A informação é, hoje em dia, um dos motores da actividade humana. De facto, independentemente do tamanho, natureza ou mesmo actividade de uma organização, a verdade é que esta precisa de informação para poder executar e prosseguir a sua missão e cumprir os objectivos a que se propõe.

De acordo com Galliers (1987:4), informação é uma colecção de dados que, quando apresentada de determinada forma e em determinado momento, melhora o conhecimento do indivíduo que a recebe, de modo a que este indivíduo se torne mais capaz de realizar a acção ou decisão a que se propõe.

A informação representa a inteligência competitiva dos negócios e é reconhecida como activo crítico para a continuidade operacional da empresa (Sêmola, 2003:160). É um processo pelo qual a empresa informa-se sobre ela própria e seu ambiente, e por ela informa ao seu ambiente sobre ela mesma (Mañas, 1999:282).

2.2.TECNOLOGIAS DE INFORMAÇÃO

A tecnologia, de uma forma ou de outra, está começando a ser um modo de conseguir vantagem competitiva em variedade de operações que antes não sofriam influência da tecnologia. A tecnologia pode, e de fato deve, para muitas das operações, contribuir para o desempenho em longo prazo das empresas (Slack, Chambers & Johnston, 2002:747).

Segundo Albertin (2002:178), TI é tudo aquilo com que se pode obter, armazenar, tratar, comunicar e disponibilizar a informação. Essa definição realça a criação das infraestruturas de TI, interna e externa à organização, e sua aplicação aos processos da cadeia de valor de negócio.

2.3.SISTEMAS DE INFORMAÇÃO

A importância das tecnologias de informação na nossa vida é sobretudo concretizada pelas funcionalidades que são implementadas ao nível do *software*, e que são disponibilizadas com o suporte de um conjunto de dispositivos diversos (*hardware*). O primeiro pode ser considerado o componente lógico dos sistemas de informação, o segundo o componente físico.

A visão mais tradicional sobre o conceito de *software* limita-se a considerá-lo como um conjunto de programas, constituído por blocos de código. Outros autores englobam ainda neste conceito a documentação de apoio que é produzida. No entanto, quando falamos actualmente do componente lógico que serve de suporte às necessidades das organizações, o conceito mais abrangente normalmente utilizado é o de sistema de informação.

Tal como em muitas outras situações no domínio da informática, não existe uma definição formal e consensual deste conceito. Segundo Da Silva (2007:172), Um SI pode ser definido como um conjunto de componentes interligados que gera, colecta, processa e

distribui informações em uma organização com alguma finalidade prática. Preferencialmente deve se utilizar de recursos tecnológicos de informática para realizar suas operações. Destina-se a fornecer elementos confiáveis de apoio à tomada de decisão, auxiliando directores, gestores e funcionários na organização, possibilitando a análise de problemas, sua visualização e soluções.

2.4. ANALISE E DESENVOLVIMENTO DE SOFTWARES

Quando se fala do desenvolvimento de *software*, são frequentemente aplicadas expressões diferentes mas que muitas vezes representam a mesma ideia. É por isso fundamental a clarificação de alguns conceitos básicos, que se encontram, em geral, associados às áreas científica/tecnológica de “sistemas de informação” e de “engenharia de software”. Estes conceitos, de natureza técnica, são frequentemente mal entendidos e/ou mal aplicados. Tais conceitos são sobretudo os de processo, metodologia e ciclo de vida.

O **processo de desenvolvimento de *software*** é um conceito de âmbito muito vasto, e pretende designar uma sequência de actividades, normalmente agrupadas em fases e tarefas, que são executadas de forma sistemática e uniformizada, que são realizadas por intervenientes com responsabilidades bem definidas, e que a partir de um conjunto de *inputs* produzem um conjunto de *outputs*. Do ponto de vista da garantia da qualidade do produto final (o *software*), é fundamental que o processo seja realizado segundo parâmetros que permitam também aferir a respectiva qualidade, isto é, não conseguiremos otimizar o resultado final sem uma preocupação no processo que o produz.

Nesta perspectiva, o conceito de **metodologia**, para além da sequência de etapas e procedimentos recomendados para serem aplicados durante o processo de desenvolvimento de sistemas de informação (ou seja, uma metodologia pressupõe a

existência de um processo), acrescenta a esta definição a utilização de um conjunto de ferramentas, técnicas e notações (Silva, A. M. & Videira, C. (2001:31).

De acordo com estas definições de processo e metodologia, o conceito de **ciclo de vida** pode ser encarado como um sinónimo de processo. A expressão ciclo de vida é mais antiga, aparecendo normalmente associada às abordagens tradicionais, enquanto o termo processo aparece sobretudo no contexto das abordagens mais recentes.

2.5.METODOLOGIAS DE DESENVOLVIMENTO DE SISTEMAS

As metodologias de sistemas são utilizadas para estabelecer ordem, definir padrões e usar técnicas já provadas no desenvolvimento de sistemas, agilizando o processo e garantindo maior qualidade no desenvolvimento. Actualmente existem dois tipos de metodologias: a estruturada e a orientada por objectos. As diferenças nas metodologias estão nas técnicas de construir o processo de negócio, as definições dos dados e os modelos de eventos.

As metodologias para desenvolvimento de sistemas devem acompanhar o ciclo de vida dos sistemas atendendo as fases de análise, desenho, construção, implementação, testes, qualidade assegurada, integração e manutenção.

2.5.1. METODOLOGIAS ESTRUTURADAS

Durante os anos 70 e 80 assistiu-se a um importante salto qualitativo com a introdução de um conjunto de metodologias que se baseavam essencialmente em técnicas estruturadas de decomposição funcional. Estas metodologias tinham por objectivo formalizar o processo de identificação de requisitos, de modo a reduzir as possibilidades de má interpretação dos mesmos e introduzir técnicas baseadas nas melhores práticas ao processo de análise e desenho. A designação de metodologias estruturadas deriva da aplicação de um conjunto

de princípios semelhante ao utilizado pelas linguagens de programação com o mesmo nome, nomeadamente o princípio da decomposição funcional.

Foi neste contexto que apareceram pela primeira vez os conceito ciclo de vida e de metodologias de desenvolvimento de *software*, com uma sequência de fases e actividades, com *inputs* e *outputs*, regras, intervenientes, técnicas, notações, ferramentas, documentação, técnicas de gestão, etc., com o objectivo de prestar mais atenção ao processo global, e menos à programação. A maioria destas metodologias adoptaram o modelo em cascata, em que cada actividade tem que ser completada e finalizada antes que a actividade seguinte possa ser iniciada.

As metodologias estruturadas propuseram diversos tipos de notações, sendo de destacar as representações gráficas e não-gráficas tais como Diagramas de fluxo de dados (DFD), Diagramas entidade Relacionamento (DER), Diagrama de transição de estados (DTE), Diagrama do ciclo de vida de entidade, Dicionários de dados, Matrizes entidade-processo e Fluxogramas.

Algumas das metodologias estruturadas mais conhecidas são: SSADM (Structured Systems Analysis and Design Methodology), STRADIS (Structured analysis, design and implementation of information Systems) e Yourdon Systems Method proposta por Edward Yourdon.

2.5.2. METODOLOGIAS ORIENTADAS POR OBJECTOS

Os métodos convencionais de desenvolvimento de sistemas (análise e projecto estruturado) apresentam algumas limitações que introduzem pontos de descontinuidade no processo de desenvolvimento de sistemas de informação. Estas limitações ocorrem principalmente por serem utilizados diagramas diferentes em cada fase do ciclo de vida de um sistema, acarretando perda de informação e inconsistências na transição de um

diagrama para outro. A maioria destas técnicas e metodologias estruturadas apresentam vários problemas, entre os quais podemos destacar:

- Não conseguem lidar adequadamente com o problema da complexidade e do tamanho crescente dos sistemas.
- Não resolvem o problema da crescente actividade de manutenção do *software*.
- Verifica-se com frequência a má compreensão dos requisitos do utilizador, por parte dos intervenientes técnicos.
- Permanece a dificuldade de lidar com alterações aos requisitos.
- A integração e reutilização de módulos e componentes do sistema não são fáceis.
- Os erros de concepção são descobertos tardiamente.
- A qualidade do *software* é baixa e o seu desempenho inadequado.
- Não é fácil identificar quem fez o quê, quando e porquê.

Os métodos orientados a objectos utilizam um modelo único, o qual é utilizado em todas as fases do ciclo de vida de um sistema. Dessa forma, o processo de desenvolvimento de sistemas é simplificado, interactivo e controlável. Cada iteração acrescenta características ao modelo, de forma a haver menores possibilidades de inconsistências e erros.

O desenvolvimento de sistemas baseado em objectos concentra a maior parte do esforço na fase de análise de requisitos. Esse esforço adicional é compensado pela implementação mais rápida e mais simples do projecto. O sistema resultante é mais completo e de fácil manutenção devido ao encapsulamento e ao reuso.

A análise e projecto orientado a objectos, se bem que em termos práticos tenha sido primeiramente concretizada ao nível das linguagens de programação, não tem impacto apenas a esse nível. O conceito da orientação por objectos (OO ou O-O, do inglês object-oriented) baseia-se de facto numa nova forma de analisar o mundo. A abordagem seguida

"reproduz" a forma como o ser humano se apercebe e expressa a realidade que o rodeia. Ele classifica e subdivide o mundo em diferentes objectos, com base nas diferenças e semelhanças existentes ao nível das características e comportamento dos mesmos objectos. As técnicas orientadas por objectos identificam e definem cada objecto de modo a reutilizá-lo, da mesma forma que o ser humano acumula conhecimento com base no previamente adquirido.

Ao longo das décadas de 1980 e de 1990 surgiram inúmeras propostas de metodologias, sobretudo concentradas nas tarefas de análise e desenho, utilizando os conceitos relacionados com o paradigma da orientação por objectos. Algumas das metodologias mais significativas, quer pela sua utilização, quer pela relevância dos conceitos abordados são, o Método *de Booch* proposto por Grady Booch em 1991, *OMT (Object Modelling Technique)* por James Rumbaugh em 1991, *OOSE (Object Oriented Software Engineering)* por Ivar Jacobson em 1992 e *OOAD (Object Oriented Analysis and Design)* proposto por Peter Coad e Edward Yourdon em 1991.

As abordagens orientadas por objectos são muito ricas em termos da terminologia utilizada. Podemos distinguir essencialmente dois grandes grupos de conceitos: (1) os princípios, que constituem um conjunto de ideias base de todo o paradigma, e alguns deles são mesmo os requisitos necessários para um sistema ser considerado orientado por objectos; (2) e as restantes noções base, comuns a todos os sistemas orientados por objectos.

No grupo dos princípios orientadores e que definem o paradigma podemos incluir os seguintes conceitos:

- **Encapsulamento da informação:** é o processo de "esconder" todos os detalhes de um objecto que não contribuem para as suas características essenciais nem para a disponibilização de funcionalidades para o seu exterior.
- **Herança:** representa a definição de relações entre classes através da qual uma subclasse partilha, acrescenta ou redefine operações e atributos a partir de uma ou mais superclasses.
- **Polimorfismo:** é a capacidade de "esconder" várias implementações distintas através de uma única interface.
- **Abstracção:** é a representação concisa duma ideia ou objecto mais complexa, incidindo sobre as características essenciais do objecto. As abordagens tradicionais concretizam esta ideia pelas abstracções funcionais (processos), enquanto os métodos orientados por objectos utilizam objectos.

Para além dos princípios que funcionam como as grandes linhas orientadoras do paradigma da orientação por objectos, existem outros conceitos que são fundamentais para a boa compreensão do mesmo, nomeadamente os conceitos de objectos e classes, para além de outros que estão directamente relacionados com estes.

O conceito básico é obviamente o de **objecto**, para o qual não existe uma definição única. Pelo contrário, praticamente cada metodologia e respectivo autor apresentam definições próprias. Resumidamente, e de forma a integrar as diversas definições, podemos dizer que os objectos representam entidades físicas, conceptuais ou apenas necessárias para a representação em computador; um objecto pode ser um conceito, uma Abstracção ou uma entidade, com fronteiras bem definidas e que tem um significado para um problema e respectiva solução; um objecto tem estado, comportamento e identidade.

Os **atributos** são propriedades nomeadas de um objecto que indicam os valores possíveis que esse atributo pode assumir ao longo do tempo; o estado de um objecto é definido pelo valor dos seus atributos.

O **comportamento** de um objecto é definido como o conjunto de acções que um objecto pode realizar de forma independente.

As **classes** são descrições de grupos de objectos com propriedades (atributos), comportamento (operações) e relações comuns. É uma das concretizações do conceito Abstracção no paradigma da orientação por objectos.

A **interface** é o conjunto de operações e atributos disponibilizados por uma classe, que consoante a visibilidade se pode dividir em três partes: pública (visível para todos os objectos do sistema); protegida (só visível pelas suas subclasses); privada (faz parte da interface mas não é visível para nenhuma outra classe do sistema, só está disponível na implementação da própria classe).

A proliferação das metodologias que tinham como base os conceitos da orientação por objectos levou ao aparecimento de diversas notações e técnicas de modelação, que muitas vezes são partilhadas entre várias delas. Os recentes esforços de unificação permitiram que algumas dessas técnicas se tenham destacado. As principais notações utilizadas estão praticamente todas presentes na metodologia da Rational que se baseia em muitos conceitos abordados nas outras metodologias, e fazem parte da linguagem de modelação UML apresentada a seguir.

2.5.2.1. MODELAGEM UML

UML é o método sucessor de diversos outros métodos de análise e projecto orientado a objectos que surgiu no final dos anos 80 e no início dos anos 90. Ele é a união entre os

métodos de Booch, Rumbaugh (OMT) e Jacobson, indo inclusive além desses métodos. (Fowler, 1999:30). Actualmente a UML é considerada pela OMG a linguagem padrão dos processos de desenvolvimento de sistemas baseados na orientação a objectos.

A **Unified Modeling Language (UML)** é uma linguagem que permite especificar, construir, visualizar e documentar os artefactos de um sistema baseado em *software*, bem como modelar negócios e outros sistemas não baseados em *software*. Ela fornece o conteúdo semântico que formaliza e orienta todo o processo de desenvolvimento. Neste sentido, a UML não é um processo, nem uma ferramenta, nem uma linguagem de programação. Seu objectivo principal é gerar a modelagem dos objectos do mundo real garantindo a interoperabilidade dos recursos envolvidos no processo de desenvolvimento.

Os diagramas são conceitos que traduzem a possibilidade de agrupar elementos básicos e suas relações de uma forma lógica ou de uma forma estrutural. Existem diferentes tipos de diagramas em UML. Em cada tipo de diagrama é usado um subconjunto dos elementos básicos previamente descritos, com diferentes tipos de relações que tenha sentido existir. Nenhum modelo é suficiente por si só. Qualquer sistema não-trivial é melhor representado através de pequeno número de modelos razoavelmente independentes. O UML define diferentes tipos de diagramas, cuja utilização e aplicação permitem dar visões complementares tais como:

- Diagramas de casos de utilização, que representam a visão do sistema na perspectiva do seu utilizador.
- Diagramas de classes que permitem especificar a estrutura estática de um sistema segundo a abordagem orientada por objectos.
- Diagramas de interacção entre objectos (diagramas de sequência e diagramas de colaboração) e diagramas de transição de estados e diagramas de actividades, que

permitem especificar a dinâmica ou o comportamento de um sistema segundo a abordagem orientada por objectos.

- Diagramas de componentes e diagramas de instalação, que dão uma visão da disposição dos componentes físicos (*software e hardware*) de um sistema.

O UML alarga o âmbito de aplicações alvo comparativamente com outros métodos existentes, designadamente porque permite, por exemplo, a modelação de sistemas concorrentes, distribuídos, para a Web, sistemas de informação geográficos, etc. A ênfase do UML é na definição de uma linguagem de modelação *standard*, e por conseguinte, o UML é independente das linguagens de programação, das ferramentas CASE, bem como dos processos de desenvolvimento. Apesar da especificação do UML incluir sugestões para os fabricantes de ferramentas a adoptarem na apresentação dessas notações (e.g., tópicos como o desenho de diagramas, cor, navegação entre esquemas, etc.), não aborda todos os requisitos necessários por não ser esse, propositadamente, o seu objectivo. O objectivo do UML é que, dependendo do tipo de projecto, da ferramenta de suporte, ou da organização envolvida, devem ser adoptados diferentes processos/metodologias, mantendo-se contudo a utilização da mesma linguagem de modelação.

CAPITULO 3 - A EMPRESA

Este capítulo apresenta a empresa OMH, o sistema actual de gestão de suas contas bancárias, suas deficiências e as melhorias que se esperam com a implantação do sistema a ser desenvolvido.

3.1. ORGANIZAÇÕES MAMADE HANIF

A OMH é uma empresa Moçambicana ou melhor, um grupo de empresas, que trabalha na área de Importações e Exportações com escritórios nas principais cidades capitais do País, tendo a sua sede na cidade de Nacala, na província de Nampula.

A OMH está no mercado há mais de duas décadas e tem como *stakeholders* companhias bastantes conhecidas como a EDM (Eletricidade de Moçambique), Africom, Merec Industries, Ferro de Moçambique, só para nomear algumas. Fazem parte da OMH, as empresas Grupo MH Merceria, OMH Electro Ferragem e a Sanal – Sacaria de Nacala.

Este grupo de empresas tem diversas contas bancárias, nas diversas cidades onde estão localizadas e em diferentes bancos. Diariamente são enviados relatórios de saldos para os proprietários da empresa, que a partir daí podem tomar decisões em relação a transacções financeiras. Montes de papéis são gastos no processo de gestão destas contas de acordo com o sistema actual em vigor, tornando difícil a obtenção de relatórios a tempo e horas.

3.2. A EMPRESA ANTES DO SISTEMA

Antes da implantação do sistema, a empresa era administrada e controlada apenas com apontamentos diários dos movimentos. Esse procedimento não disponibilizava informações no momento em que elas fossem realmente necessárias. Se, por acaso, o proprietário ligasse pedindo o saldo de uma conta, ou ainda, os movimentos durante um

certo período de tempo, o funcionário precisaria procurar muito e ainda fazer alguns cálculos para informá-lo.

Este sistema manual de funcionamento apresenta muitas falhas, com principal ênfase para a inconsistência dos dados disponíveis e as demasiadas “*ginásticas*” para a geração de relatórios, factor que influenciava negativamente o rendimento das tarefas e o processo de tomada decisão. Isso se devia, não só pelo número elevado de contas, mas também pelo fato de as informações não possuírem uma fonte única e de fácil acesso. Nem sempre se podia confiar no extracto do banco, uma vez que este poderia não reflectir alguns movimentos que não foram valorizados, como por exemplo cheques emitidos que ainda não tenham sido cobrados. Como consequência, inconsistências eram geradas onde a única saída viável era o levantamento de todas as informações.

3.3.A EMPRESA DEPOIS DO SISTEMA

Após a implantação do sistema, os apontamentos diários não deixarão de ocorrer como ocorriam no passado. Porém, agora existe a facilidade na obtenção e disponibilização das informações para consulta e esclarecimento dos proprietários da empresa.

Logo após a obtenção dos dados diários na empresa, o funcionário sentará a frente de um computador e fará lançamento dos movimentos no sistema bancOMH. Essas informações tornam-se disponíveis imediatamente, possibilitando que todos usuários tenham acesso, gerem relatórios, e acompanhem com maior detalhe as operações que estão ocorrendo com relação as contas bancárias da empresa.

CAPITULO 4 - REQUISITOS E MODELAGEM

Para facilitar a visualização do bancOMH e ajudar na compreensão de como as estruturas internas do mesmo se relacionam, foram elaborados diagramas usando modelagem UML feita através da abordagem orientada a objectos. O UML é uma linguagem diagramática, utilizável para especificação, visualização e documentação de sistemas de *software* que actualmente carrega o estatuto de norma no âmbito da OMG e da ISO, tendo vindo a ser adoptado progressivamente pela indústria em todo o mundo. A seguir são apresentados os requisitos do sistema e os diagramas desenvolvidos na fase de análise e desenho do sistema.

4.1.REQUISITOS DO SISTEMA

Requisitos são objectivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema. Os requisitos de *software* são, obviamente, aqueles dentre os requisitos de sistema que dizem respeito a propriedades do *software*. Tradicionalmente, os requisitos de *software* são separados em requisitos funcionais e não-funcionais.

4.1.1. Requisitos funcionais

Os requisitos funcionais são a descrição das diversas funções que clientes e usuários querem ou precisam que o *software* ofereça. Eles definem a funcionalidade desejada do *software*. O termo função é usado no sentido genérico de operação que pode ser realizada pelo sistema, seja através comandos dos usuários ou seja pela ocorrência de eventos internos ou externos ao sistema. A especificação de um requisito funcional deve determinar *o que* se espera que o *software* faça, sem a preocupação de *como* ele faz.

Os requisitos funcionais para o bancOMH são:

- O sistema deve separar os débitos dos créditos
- Todos os lançamentos de movimentos devem estar ordenados de acordo com a data por ordem crescente
- O sistema deve emitir diariamente relatórios dos saldos de todas as contas separadas por cidades
- Os usuários podem pesquisar saldos de datas anteriores
- O sistema deve permitir a busca detalhada dos movimentos de acordo com vários critérios, útil para encontrar um determinado movimento ou grupo de movimentos
- Todas as contas e usuários devem estar associados a uma única cidade
- Todas operações devem guardar os dados do usuário que a executou
- O sistema deve possibilitar a emissão de extractos durante um determinado período de tempo

4.1.2. Requisitos não-funcionais

Requisitos não-funcionais são as qualidades globais de um *software*, como manutenibilidade, usabilidade, desempenho, custos e várias outras. Normalmente estes requisitos são descritos de maneira informal, de maneira controversa (por exemplo, o proprietário quer segurança mas os usuários querem facilidade de uso) e são difíceis de validar.

Os requisitos não-funcionais do sistema são:

- Só o administrador pode adicionar e eliminar contas bancarias
- O sistema deve garantir que os dados estão protegidos de acessos não autorizados.
- Todos os menus principais devem ter atalhos para fácil uso
- O sistema deve separar os usuários em níveis de acesso

- O sistema deve ordenar os movimentos de acordo com as datas e da mesma forma, o saldo da conta deve obedecer à ordem
- Nas telas de lançamento de movimentos deve ter *links* para outros tipos de movimentos
- O *software* deve ser operacionalizado nos sistemas operativos *Windows*
- Todos usuários podem gerar relatórios
- Todas as telas do sistema devem ser leves

4.2. DIAGRAMAS DE MODELAGEM

4.2.1. DIAGRAMA DE CASOS DE USO

Um caso de uso é uma sequência de acções que um ou mais actores realizam num sistema de modo a obterem um resultado particular. Este é um diagrama que permite dar uma visão global e de alto nível do sistema, sendo fundamental a definição correta da sua fronteira. A figura abaixo apresenta o diagrama de casos de uso do bancOMH que representa a visão do sistema na perspectiva do seu utilizador.

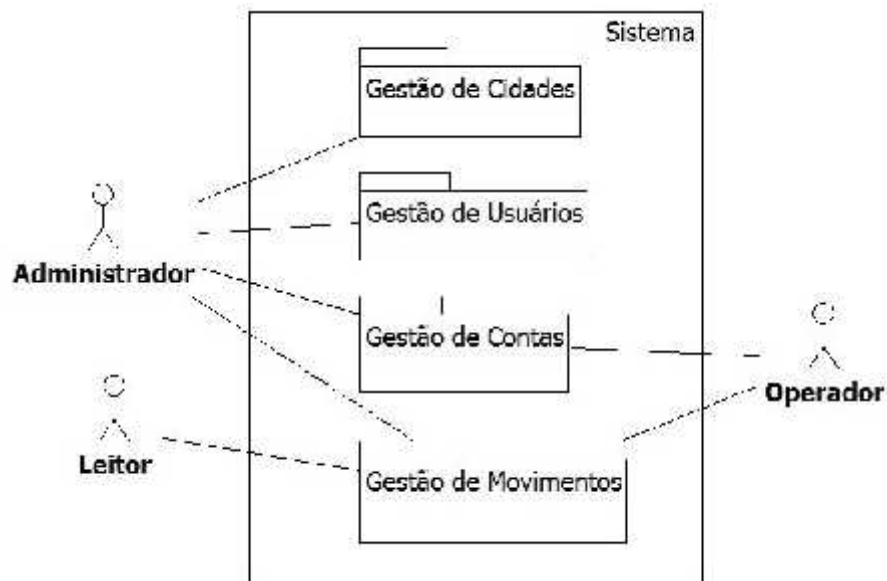


Figura 1 - Diagrama de Casos de Uso

Os elementos do diagrama acima demonstrado foram agregados em grupos de forma que semântica ou estruturalmente façam sentido. Estes grupos são chamados de **pacotes** (*packages*) e são elementos meramente organizacionais. Estes pacotes apresentam atributos semelhantes e foram chamados de *Gestão* pois fazem uso das propriedades CRUD (*Create, Read, Update e Delete*) das classes ou seja, adicionar, ler, actualizar e eliminar. As figuras posteriores apresentam em detalhe os *pacotes* em questão.



Figura 2 - Pacote de Gestão de Cidades



Figura 3- Pacote de Gestão de Usuários



Figura 4 - Pacote de Gestão de Contas

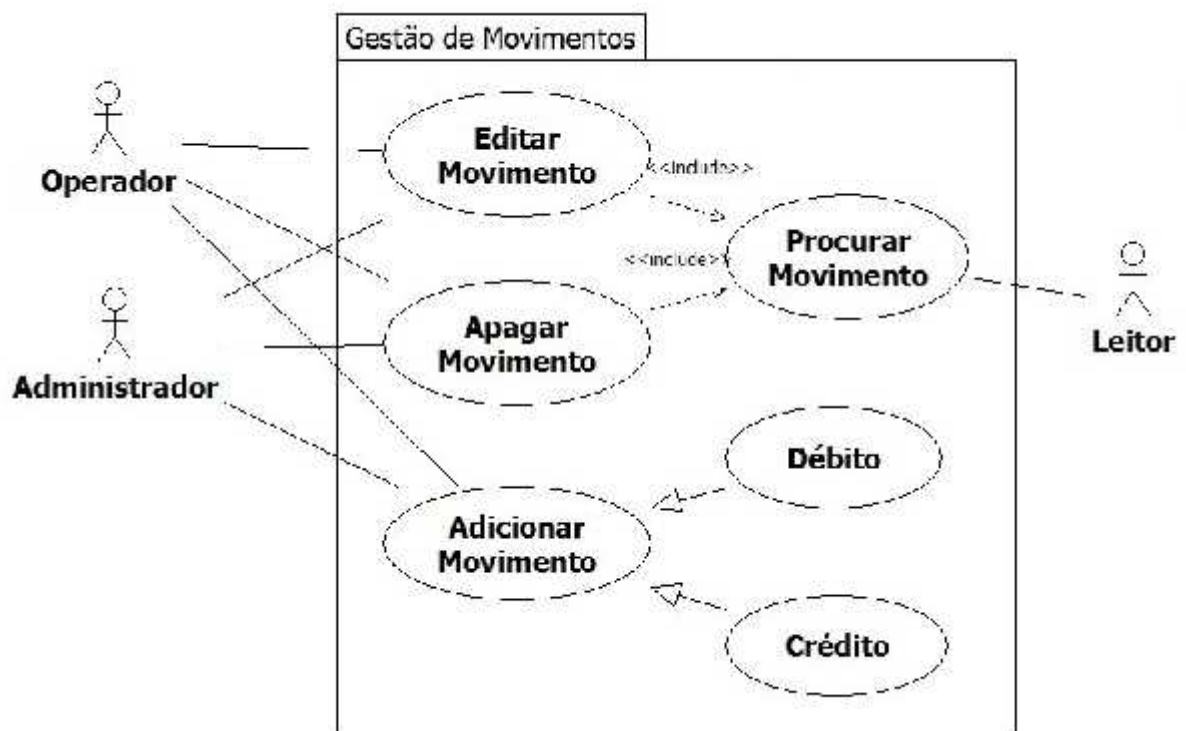


Figura 5 - Pacote de Gestão de Movimentos

Existem três tipos de actores que interagem com o sistema sendo eles: Administradores, operadores e leitores. Os administradores têm total acesso e controlo sobre o sistema, são eles que criam os outros usuários, eliminam contas e tratam da parte funcional do sistema.

Os operadores têm acesso apenas às contas para a cidade que estão registados e podem criar movimentos, alterar, porem não podem eliminar as contas. Os operadores são os usuários que mais usarão o sistema, sendo estes os maiores responsáveis pela introdução dos dados no sistema.

Os leitores são aqueles que só podem verificar relatórios, dados relacionados com as contas e movimentos sem no entanto poder alterar ou eliminar os mesmos. Todos podem gerar relatórios, efectuar leituras no sistema existindo apenas limitação para os operadores que só podem ter acesso aos dados das contas para a sua cidade. O actor leitor foi criado tendo em mente os proprietários da empresa que são os principais responsáveis pelas tomadas de decisão.

A seguir é apresentada a documentação dos casos de uso do sistema. Esta descrição textual contem o fluxo principal das operações bem como os fluxos alternativos caso existam.

| | |
|---------------------------------|--|
| Nome do caso de Uso | Adicionar Cidade |
| Descrição | Permite adicionar novas cidades ao sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. |
| Pós-Condições | Cidade adicionada. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Solicita cidade} |
| | Solicita o nome da cidade. (A1) |
| Fornece o nome da cidade. | |
| | {Valida cidade} |
| | Avalia se a cidade já existe na base de dados. (A2) |
| | {Adiciona cidade} |
| | Adiciona a cidade na base de dados. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista das cidades. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Solicita cidade} | O administrador pode decidir encerrar o caso de uso sem fornecer a cidade. Nesse caso retomar o fluxo básico em {Fim} |
| A2: em {Valida cidade} | Caso o administrador introduza uma cidade que já existe na base de dados então o sistema mostra uma informação na tela e retoma o fluxo básico em {Solicita cidade} |

Tabela 1 - Descrição do Caso de Uso Adicionar Cidade

| | |
|--|--|
| Nome do caso de Uso | Editar cidade |
| Descrição | Permite editar as cidades existentes no sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. Cidade existente. |
| Pós-Condições | Cidade actualizada. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Mostra cidades} |
| | O sistema mostra a lista das cidades existentes. |
| O administrador selecciona a cidade que deseja editar. | |
| Fornece os novos dados. | |
| | {Valida dados} |
| | Avalia se a cidade existe ou não. (A1) |
| | {Actualiza cidade} |
| | Actualiza a cidade na base de dados. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista das cidades. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida dados} | Caso o administrador introduza uma cidade que já existe na base de dados então o sistema mostra uma caixa de mensagem de erro na tela e retoma o fluxo básico em {Mostra cidades} |

Tabela 2 - Descrição do Caso de Uso Editar Cidade

| | |
|--|---|
| Nome do caso de Uso | Apagar Cidade |
| Descrição | Permite apagar as cidades existentes no sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. Cidade existente. |
| Pós-Condições | Cidade eliminada. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Mostra cidades} |
| | O sistema mostra a lista das cidades existentes. |
| Selecciona a cidade que deseja eliminar. | |
| | {Valida operação} |
| | Pergunta se o administrador deseja apagar a cidade. |
| Fornece resposta. (A1) | |
| | {Elimina cidade} |
| | Elimina a cidade seleccionada. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista das cidades. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida operação} | Caso o administrador seleccione a resposta negativa então nesse caso retomar o fluxo básico em {Fim} |

Tabela 3 - Descrição do Caso de Uso Apagar Cidade

| | |
|----------------------------------|--|
| Nome do caso de Uso | Adicionar Usuário |
| Descrição | Permite adicionar usuários ao sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. |
| Pós-Condições | Usuário adicionado. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Solicita usuário} |
| | Solicita os dados do usuário. (A1) |
| Fornece os dados. | |
| | {Valida usuário} |
| | Avalia se o usuário já existe na base de dados. Verifica se o usuário foi associado a uma cidade existente. (A2) |
| | {Adiciona usuário} |
| | Adiciona o usuário na base de dados. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista dos usuários. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Solicita usuário} | O administrador pode decidir encerrar o caso de uso sem fornecer o usuário. Nesse caso retomar o fluxo básico em {Fim} |
| A2: em {Valida usuário} | Caso o administrador introduza um usuário que já existe na base de dados, ou introduza um usuário sem associa-lo a nenhuma cidade então o sistema mostra uma informação na tela e retoma o fluxo básico em {Solicita usuário} |

Tabela 4 - Descrição do Caso de Uso Adicionar Usuário

| | |
|---|---|
| Nome do caso de Uso | Editar usuário |
| Descrição | Permite editar os usuários existentes no sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. Usuário existente. |
| Pós-Condições | Usuário actualizado. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Mostra usuários} |
| | O sistema mostra a lista dos usuários existentes. |
| O administrador selecciona o usuário que deseja editar. | |
| Fornece os novos dados. | |
| | {Valida dados} |
| | Avalia se o usuário já existe na base de dados. (A1) |
| | {Actualiza usuário} |
| | Actualiza o usuário na base de dados. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista dos usuários. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida dados} | Caso o administrador introduza um usuário que já existe na base de dados então o sistema mostra uma caixa de mensagem de erro na tela e retoma o fluxo básico em {Mostra usuários} |

Tabela 5 - Descrição do Caso de Uso Editar Usuário

| | |
|---|---|
| Nome do caso de Uso | Apagar Usuário |
| Descrição | Permite apagar os usuários existentes no sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. Usuário existente. |
| Pós-Condições | Usuário eliminado. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Mostra usuários} |
| | O sistema mostra a lista dos usuários existentes. |
| Selecciona o usuário que deseja eliminar. | |
| | {Valida operação} |
| | Pergunta se o administrador deseja apagar o usuário. |
| Fornece resposta. (A1) | |
| | {Elimina usuário} |
| | Elimina o usuário seleccionado. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista dos usuários. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida operação} | Caso o administrador seleccione a resposta negativa então nesse caso retomar o fluxo básico em {Fim} |

Tabela 6 - Descrição do Caso de Uso Apagar Usuário

| | |
|---|--|
| Nome do caso de Uso | Adicionar Conta |
| Descrição | Permite adicionar contas ao sistema. |
| Actor envolvido | Administrador, Operador |
| Pré-condições | Sessão iniciada. |
| Pós-Condições | Conta adicionada. |
| Fluxo Básico | |
| Administrador, Operador | Sistema |
| | {Solicita dados} |
| | Solicita os dados da conta. (A1) |
| Fornece os dados. (O operador só pode criar uma conta para a cidade que está associado) | |
| | {Valida dados} |
| | Verifica se a conta foi associada a uma cidade existente. (A2) |
| | {Adiciona conta} |
| | Adiciona a conta na base de dados. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista das contas. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em{ Solicita dados } | O actor pode decidir encerrar o caso de uso sem fornecer a conta. Nesse caso retomar o fluxo básico em {Fim} |
| A2: em {Valida dados} | Caso o actor introduza uma conta que já existe na base de dados, ou introduza uma conta sem associa-la a nenhuma cidade então o sistema mostra uma informação na tela e retoma o fluxo básico em {Solicita dados} |

Tabela 7 - Descrição do Caso de Uso Adicionar Conta

| | |
|---|--|
| Nome do caso de Uso | Editar conta |
| Descrição | Permite editar as contas existentes no sistema. |
| Actor envolvido | Administrador, Operador |
| Pré-condições | Sessão iniciada. Conta existente. (Para o caso do operador, só pode editar as contas da sua cidade). |
| Pós-Condições | Conta actualizada. |
| Fluxo Básico | |
| Administrador, Operador | Sistema |
| | { Mostra contas } |
| | O sistema mostra a lista das contas existentes. Para os operadores, só as contas da cidade associada serão mostradas. |
| O actor selecciona a conta que deseja editar. | |
| Fornece os novos dados. | |
| | { Valida dados } |
| | Avalia os dados. (A1) |
| | { Actualiza conta } |
| | Actualiza a conta na base de dados. |
| | { Mostra o resultado } |
| | Informa o resultado da operação e actualiza a lista das contas. |
| | { Fim } Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em { Valida dados } | Caso o actor introduza uma conta que já existe na base de dados então o sistema mostra uma caixa de mensagem de erro na tela e retoma o fluxo básico em { Mostra contas } |

Tabela 8 - Descrição do Caso de Uso Editar Conta

| | |
|---|---|
| Nome do caso de Uso | Apagar Conta |
| Descrição | Permite apagar as contas existentes no sistema. |
| Actor envolvido | Administrador |
| Pré-condições | Sessão iniciada. Conta existente. |
| Pós-Condições | Conta eliminada. |
| Fluxo Básico | |
| Administrador | Sistema |
| | {Mostra contas} |
| | O sistema mostra a lista das contas existentes. |
| Selecciona a conta que deseja eliminar. | |
| | {Valida operação} |
| | Verifica se a conta tem movimentos ou não. Pergunta se o administrador deseja apagar a conta. |
| Fornece resposta. (A1) | |
| | {Elimina conta} |
| | Elimina a conta seleccionada. |
| | Elimina os movimentos da conta. |
| | {Mostra o resultado} |
| | Informa o resultado da operação e actualiza a lista das contas. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida operação} | Caso o administrador seleccione a resposta negativa então nesse caso retomar o fluxo básico em {Fim} |

Tabela 9 - Descrição do Caso de Uso Apagar Conta

| | |
|--------------------------------|--|
| Nome do caso de Uso | Adicionar Movimento |
| Descrição | Permite adicionar movimentos às contas existentes no sistema. |
| Actor envolvido | Administrador, Operador |
| Pré-condições | Sessão iniciada. Conta existente. |
| Pós-Condições | Movimento adicionado ou operação cancelada pelo actor. |
| Fluxo Básico | |
| Administrador, Operador | Sistema |
| | {Solicita dados} |
| | Solicita a conta do movimento. (A1) |
| Fornece a conta. | |
| | Solicita o tipo de movimento |
| Fornece o tipo de operação. | |
| | Solicita o valor da operação |
| Fornece o valor da operação | |
| | {Valida dados} |
| | Avalia os dados introduzidos (A2) |
| | {Adiciona movimento} |
| | Adiciona o movimento na base de dados. |
| | Altera o saldo da conta. |
| | {Mostra o resultado} |
| | Informa o resultado da operação. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Solicita dados} | O actor pode decidir encerrar o caso de uso sem fornecer os dados da operação. Nesse caso retomar o fluxo básico em {Fim} |
| A2: em {Valida dados} | Caso os dados introduzidos sejam inválidos então o sistema mostra uma e retoma o fluxo básico em {Solicita dados} |

Tabela 10 - Descrição do Caso de Uso Adicionar Movimento

| | |
|---|---|
| Nome do caso de Uso | Editar movimento |
| Descrição | Permite editar os movimentos existentes no sistema. |
| Actor envolvido | Administrador, Operador |
| Pré-condições | Sessão iniciada. Conta existente. (Para o caso do operador, só pode editar as contas da cidade a qual esta associado). Movimento existente. |
| Pós-Condições | Movimento actualizado ou operação cancelada pelo actor. |
| Fluxo Básico | |
| Administrador, Operador | Sistema |
| | {Mostra movimentos} |
| | O sistema mostra a lista dos movimentos existentes. Para os operadores, só as contas da cidade associada serão mostradas. |
| O actor selecciona o movimento que deseja editar. | |
| | {Solicita dados} |
| Fornece os novos dados. | |
| | {Valida dados} |
| | Avalia os dados. (A1) |
| | {Actualiza movimento} |
| | Actualiza os dados do movimento. |
| | Actualiza o saldo da conta. |
| | {Mostra o resultado} |
| | Informa o resultado da operação. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida dados} | Caso os dados introduzidos sejam inválidos então o sistema mostra uma informação na tela e retoma o fluxo básico em {Solicita dados} |

Tabela 11 - Descrição do Caso de Uso Editar Movimento

| | |
|---|---|
| Nome do caso de Uso | Apagar movimento |
| Descrição | Permite apagar os movimentos existentes no sistema. |
| Actor envolvido | Administrador, Operador |
| Pré-condições | Sessão iniciada. Conta existente. (Para o caso do operador, só pode editar as contas da cidade a qual esta associado). Movimento existente. |
| Pós-Condições | Movimento eliminado ou operação cancelada pelo actor. |
| Fluxo Básico | |
| Administrador, Operador | Sistema |
| | {Mostra movimentos} |
| | O sistema mostra a lista dos movimentos existentes. Para os operadores, só as contas da cidade associada serão mostradas. |
| O actor selecciona o movimento que deseja apagar. | |
| | {Valida operação} |
| | Pergunta se o actor deseja eliminar o movimento |
| Fornece resposta. (A1) | |
| | {Elimina movimento} |
| | Elimina o movimento seleccionado. |
| | {Actualiza conta} |
| | Actualiza o saldo da conta. |
| | Informa o resultado da operação. |
| | {Fim} Fim do caso de uso. |
| Fluxos alternativos | |
| A1: em {Valida dados} | Caso a resposta seja não então o sistema retoma o fluxo básico em {Mostra movimentos} |

Tabela 12 - Descrição do Caso de Uso Apagar Movimento

4.2.2. DIAGRAMA DE CLASSES

Um diagrama de classes ilustra um conjunto de classes, interfaces, colaborações e respectivas relações, em geral de dependência, generalização e de associação. Os diagramas de classes são usados para modelar a estrutura de um sistema. Estes modelos são também designados por “vista do desenho estático do sistema” e são usados tipicamente em três situações: (1) para modelar o vocabulário de um sistema; (2) para modelar colaborações simples; e (3) para modelar o desenho de um esquema de uma base de dados. Visa permitir a visualização das classes que compõem o sistema junto com os respectivos atributos e métodos, bem como mostrar como as classes se relacionam, complementam e transmitem informações entre si.

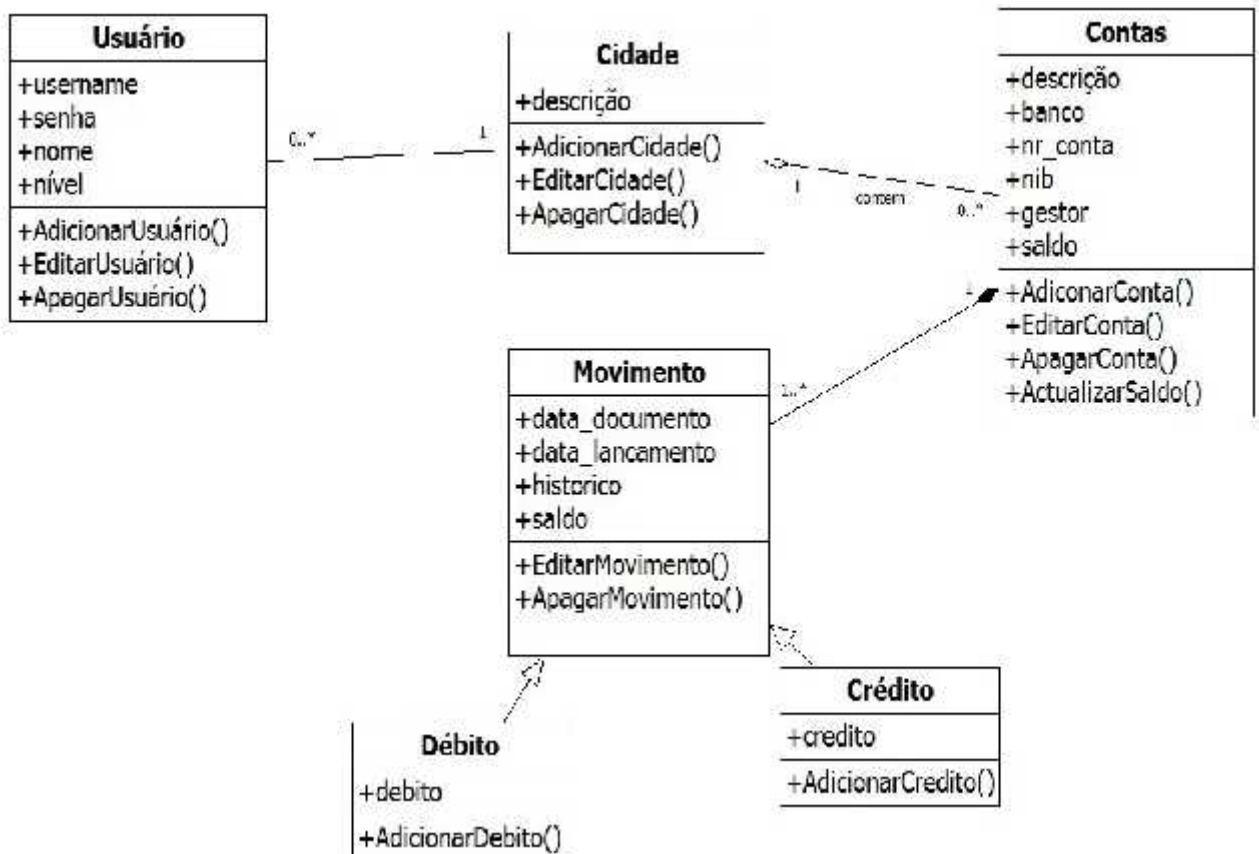


Figura 6 – Diagrama de Classes

Para melhor descrever a distribuição dos dados armazenados no sistema, é apresentado, conforme ilustra a abaixo, um diagrama que mostra o relacionamento das tabelas da base de dados. As chaves primárias estão sinalizadas com o ícone que representa uma chave e logo abaixo dela antes da linha horizontal de cor cinza estão as chaves estrangeiras da tabela, caso existam.

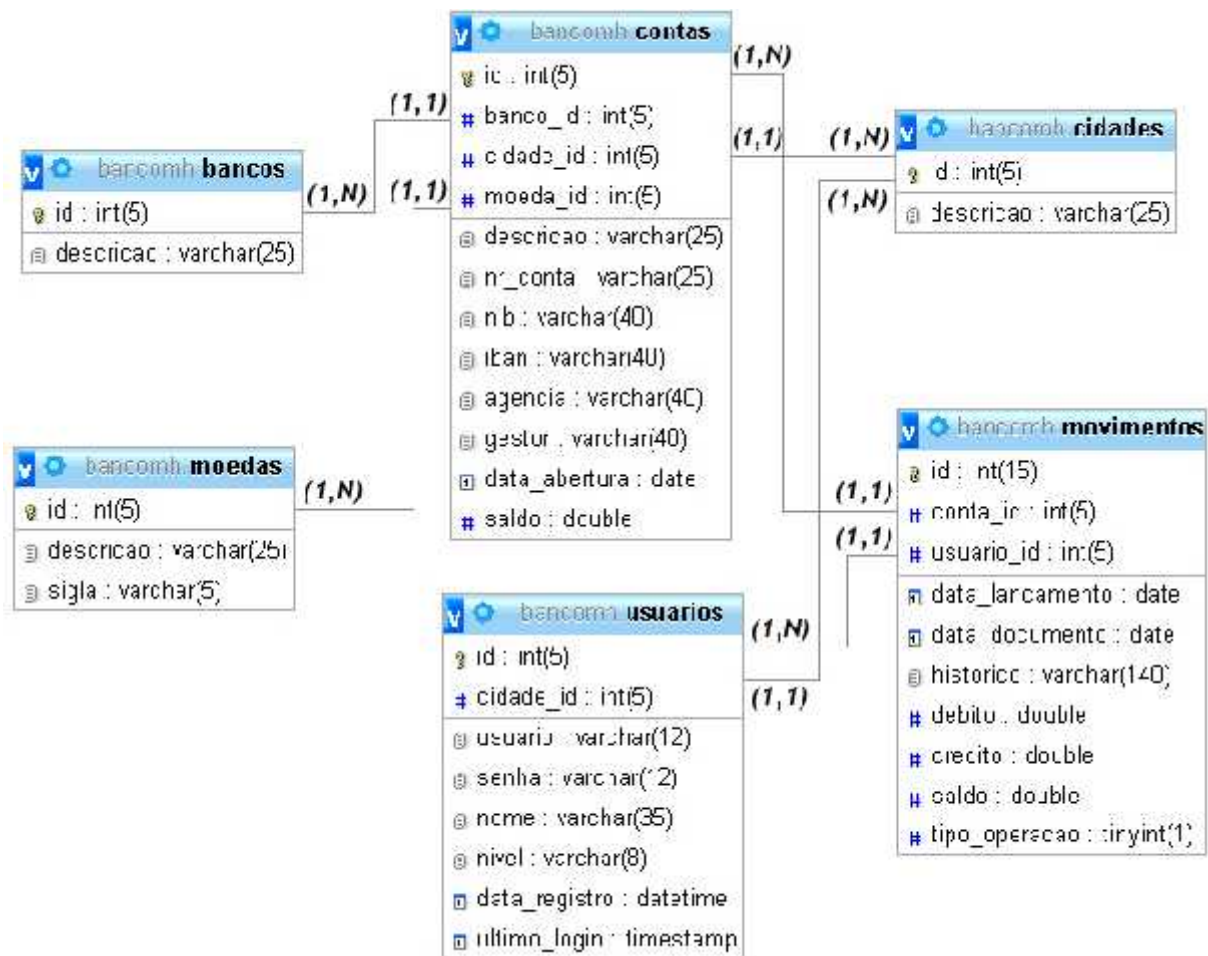


Figura 7 – Modelo Entidade-Relacionamento

4.2.3. DIAGRAMA DE SEQUÊNCIA

Um **diagrama de sequência** ilustra interações entre objectos num determinado período de tempo. É um diagrama de interacção com ênfase na ordenação temporal das mensagens trocadas entre os objectos. Em particular, os objectos são representados pelas suas “linhas de vida” ou dimensão horizontal e interagem por troca de mensagens ao longo de um determinado período de tempo, a dimensão vertical.

A figura a seguir apresenta o diagrama de sequência para o processo de introdução de um movimento no sistema. Esta operação afecta não só a entidade **movimento** mas também o objecto **conta** que deve ter o seu saldo actualizado uma vez concluída a operação.

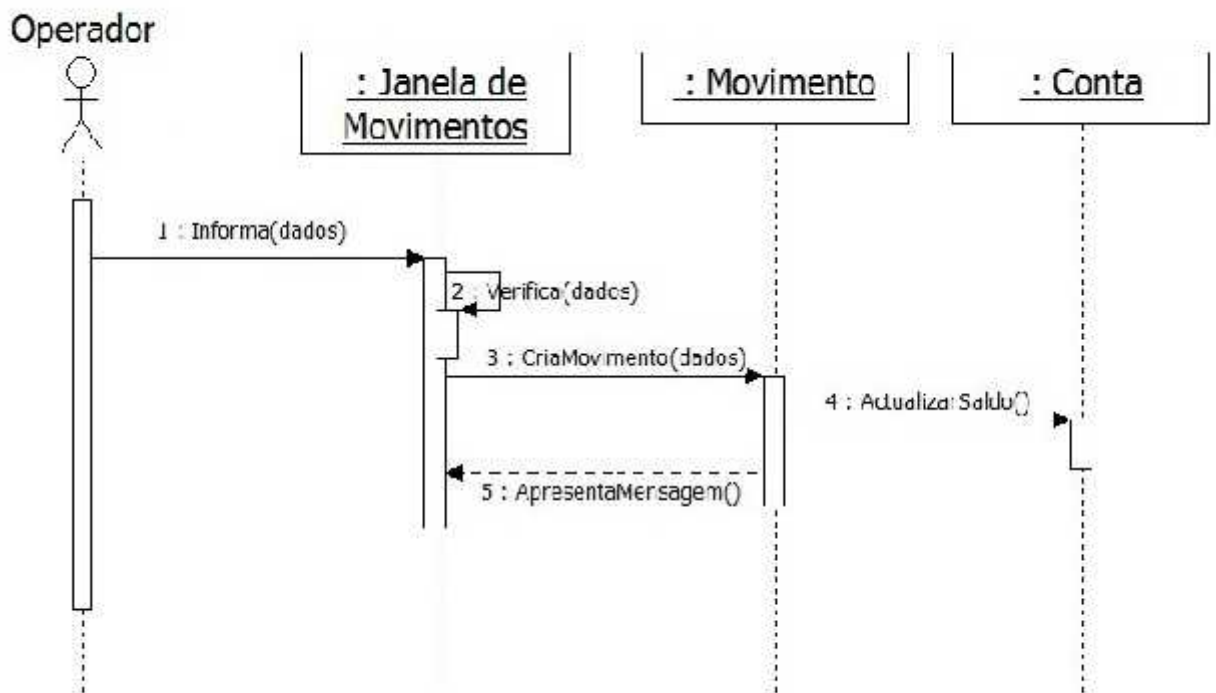


Figura 8 – Diagrama de Sequência para Criar Movimento

CAPITULO 5 - O SISTEMA

Neste capítulo estão descritos os módulos do sistema e suas respectivas transacções, dando informações mais detalhadas de como funcionam cada uma delas. Todos dados que aparecem neste capítulo são fictícios, usados meramente para fins demonstrativos.

Todas as telas do bancOMH foram feitas de modo a serem leves e livres de complicação. Na grande parte dos campos de entrada de dados, optou-se pela existência de um *ComboBox*¹ já contendo as entradas possíveis. Dessa forma impede-se que o usuário acabe fornecendo algum valor incorrecto. Os campos obrigatórios possuem procedimentos para verificação dos valores introduzidos. Os demais ficam da responsabilidade do usuário.

Cada módulo é composto de basicamente quatro transacções visíveis: criação, exibição, alteração e eliminação. A transacção de criação fornece ao usuário uma forma de alimentar as tabelas que compõem o bancOMH. A exibição dos dados registados é feita através da transacção própria de cada módulo. No geral, elas possuem formato de relatório simplificado. Telas mais complexas vêm acompanhadas de campos de filtragem de dados.

O bancOMH tem algumas transacções que, para serem realizadas, precisam de que sejam verificadas algumas premissas. O módulo dos movimentos é um excelente exemplo, não se podem fazer lançamentos sem contas registadas no sistema.

¹*ComboBox* (também *combo*) ou Caixa de Listagem como é comumente traduzido, é um controle que permite ao usuário seleccionar uma opção dentre várias de uma lista.

A acessibilidade e desempenho foram factores levados em consideração na construção das telas. Todos os campos das telas estão organizados em sequência, ou seja o usuário pode usar a tecla *Tab* para mover de um campo para outro. Em todas as telas a tecla *Esc* (*Escape*) é usada para fechar a tela e passar para a tela principal. A tecla *Enter* está programada para funcionar de acordo com o contexto da operação, seja ela criação ou alteração basta que se clique a tecla *Enter* para guardar os dados. Foram também criados *shortcuts* (atalhos) para a maioria dos itens do menu. Isto garante uma maior flexibilidade ao usuário que não precisará de usar o *mouse* para trocar de campos.

5.1.ACESSO AO BANCOMH

Ao aceder a aplicação, o usuário é confrontado com uma de tela de login precedida por um pequeno *splashform* (tela de carregamento), conforme ilustram as Figuras 9 e 10.



Figura 7 - Tela de Carregamento

Para que a autenticação seja efectuada com sucesso, deverão ser informados o nome do usuário e a sua senha. O método de autenticação é simples, sendo feita uma conexão a base de dados para a verificação dos dados introduzidos. O usuário tem apenas três tentativas de login, sendo que quando a terceira é atingida uma mensagem de erro é mostrada e o sistema é fechado automaticamente.



Figura 8– Tela de Login

5.1.1. Tela Principal

A partir do momento que o usuário consegue o acesso ao sistema, ele é conduzido a uma tela que exibe o menu principal do sistema (ver Figura 11). Alguns itens do menu principal ficam desabilitados dependendo do nível do usuário que esta acedendo o sistema.

Na parte superior temos uma barra de ferramentas que contem atalhos para os módulos principais do sistema, uma forma rápida de aceder aos módulos mais usados pelos operadores. Nesta mesma tela são encontradas uma mensagem de boas vindas, informações relacionadas com o tempo da sessão actual, bem como a data do último login efectuado pelo usuário. No canto inferior direito temos um atalho para abrir a calculadora do Windows.

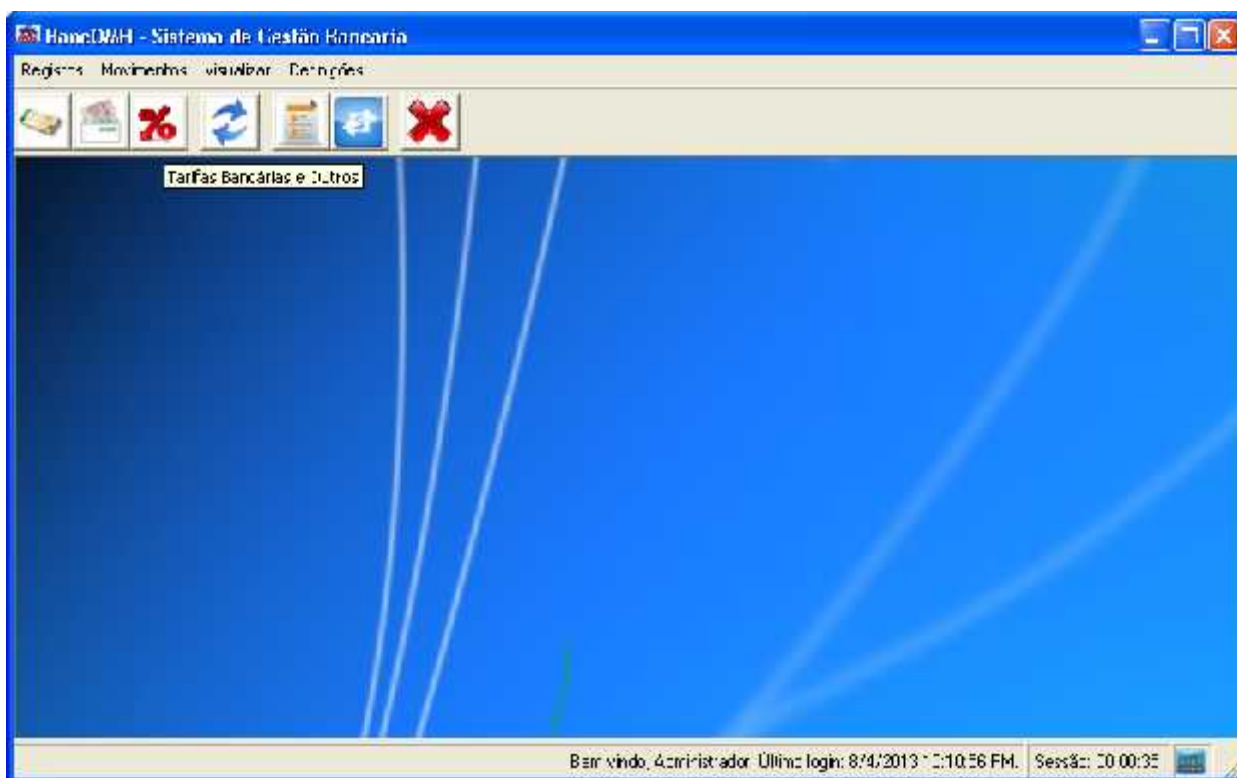


Figura 9– Tela Principal do bancOMH

5.2.MÓDULOS DE GESTÃO

5.2.1. Gestão de Cidades

Esta tela só pode ser acedida pelo administrador do sistema. Nesta tela são realizadas as operações de criação, alteração e eliminação das cidades conforme ilustra a Figura 12.



Figura 10– Tela das Cidades

As transacções descritas a seguir acontecem em praticamente todos os módulos do menu de *Gestão* sendo mencionado caso se verifique o contrário.

5.2.1.1. Exibição

A exibição é feita automaticamente a partir do momento que a tela é carregada. É feita uma conexão com a base de dados para leitura dos dados que depois aparecem na lista das cidades. Este procedimento é sempre executado quando é feita alguma alteração que necessite actualização do sistema.

5.2.1.2. Adicionar

No processo de adição de uma cidade o usuário introduz os dados no campo descrição e depois clica no botão de comando *adicionar*. Estes dados são verificados para evitar que se introduza uma mesma cidade varias vezes, e caso sejam validos, uma conexão com a base de dados é criada com a finalidade de adicionar os dados introduzidos. Feito isso, a lista das cidades é actualizada para incluir a ultima cidade adicionada.

5.2.1.3. Modificar

Para modificar uma cidade, o usuário selecciona a cidade que deseja alterar, e os dados da mesma aparecem no campo descrição. O usuário depois introduz os novos valores e clica no botão modificar que guarda as alterações efectuadas.

5.2.1.4. Eliminar

Para eliminar uma cidade o usuário deve seleccionar a cidade que deseja eliminar e depois clicar no botão *eliminar*. Uma caixa de diálogo aparece perguntando ao usuário se deseja efectuar a operação conforme ilustra a Figura 13. Caso o usuário confirme então a cidade é eliminada do sistema e uma mensagem é mostrada na tela indicando que a operação foi concluída com sucesso.

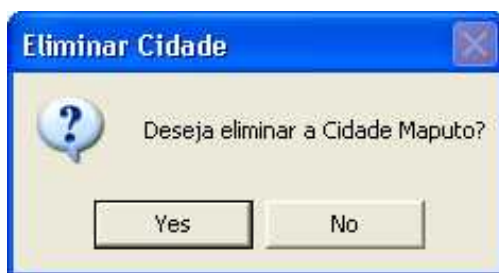


Figura 11– Confirmação de Eliminação de Cidade

No processo de eliminação existe um procedimento que verifica se existe alguma conta associada a cidade que se deseja eliminar. Caso exista, então uma mensagem é mostrada indicando que não se pode eliminar a cidade seleccionada porque uma ou mais contas estão associadas a esta (ver Figura 14).

O idealizador do sistema sugeriu que quando se verifica-se esta fase, uma caixa de diálogo deveria aparecer perguntando se o usuário deseja eliminar também todas as contas associadas a esta cidade e os respectivos movimentos. Este assunto mereceu muita atenção por parte do desenhador do sistema e os futuros usuários, pois bastava um clique errado para apagar montes de dados. Sendo assim, fez-se entender ao idealizador do mesmo que esta opção poderia ser maliciosa para o sistema. Após esta conversa foi decidido que seria melhor a exclusão desta opção.

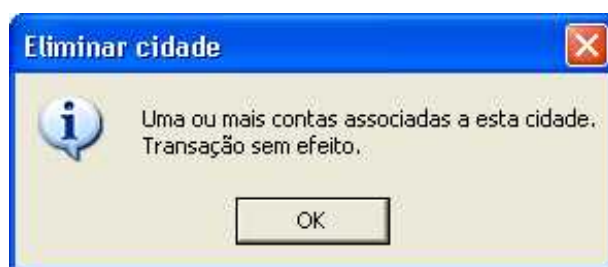


Figura 12– Mensagem de eliminação sem efeito

5.2.2. Gestão de Usuários

Semelhante a tela das cidades, só o administrador pode aceder a tela dos usuários. Conforme a visão do idealizador do sistema, existem três tipos de usuários nomeadamente

administradores, operadores e leitores. Dependendo do nível que eles estão associados, eles têm algumas restrições. O administrador tem acesso completo ao sistema. Todos usuários são associados a uma cidade e o operador tem acesso as transações relacionadas com a cidade que esta associado. O leitor só tem acesso aos relatórios e busca de movimentos. Este último nível foi criado tendo em mente os proprietários da empresa. Eles podem controlar todos lançamentos mas não podem altera-los, evitando assim que por algum motivo e falta de formação descontrolarem o sistema. Os usuários são únicos, não pode existir dois usuários com os mesmos dados de entrada. O sistema verifica esta condição quando se está a adicionar usuários.

Visto que o sistema pode ter inúmeros usuários, criou-se um campo de busca para facilitar aos administradores no processo de busca e selecção de usuários. Este campo filtra o nome do usuário, o *username* e o nível de acesso. Desde o momento que o administrador escreva leitor, aparecerão na tabela, todos os usuários com nível de acesso atribuído leitor.

A tela tem um botão de comando que permite passar para a tela de criação de cidades e voltar automaticamente a tela de usuários com a nova cidade adicionada seleccionada como a cidade do usuário a criar ou editar. Isto serve para evitar que o administrador tenha que ir a tela específica das cidades através do menu de gestão e depois voltar a tela dos usuários (ver Figura 15).

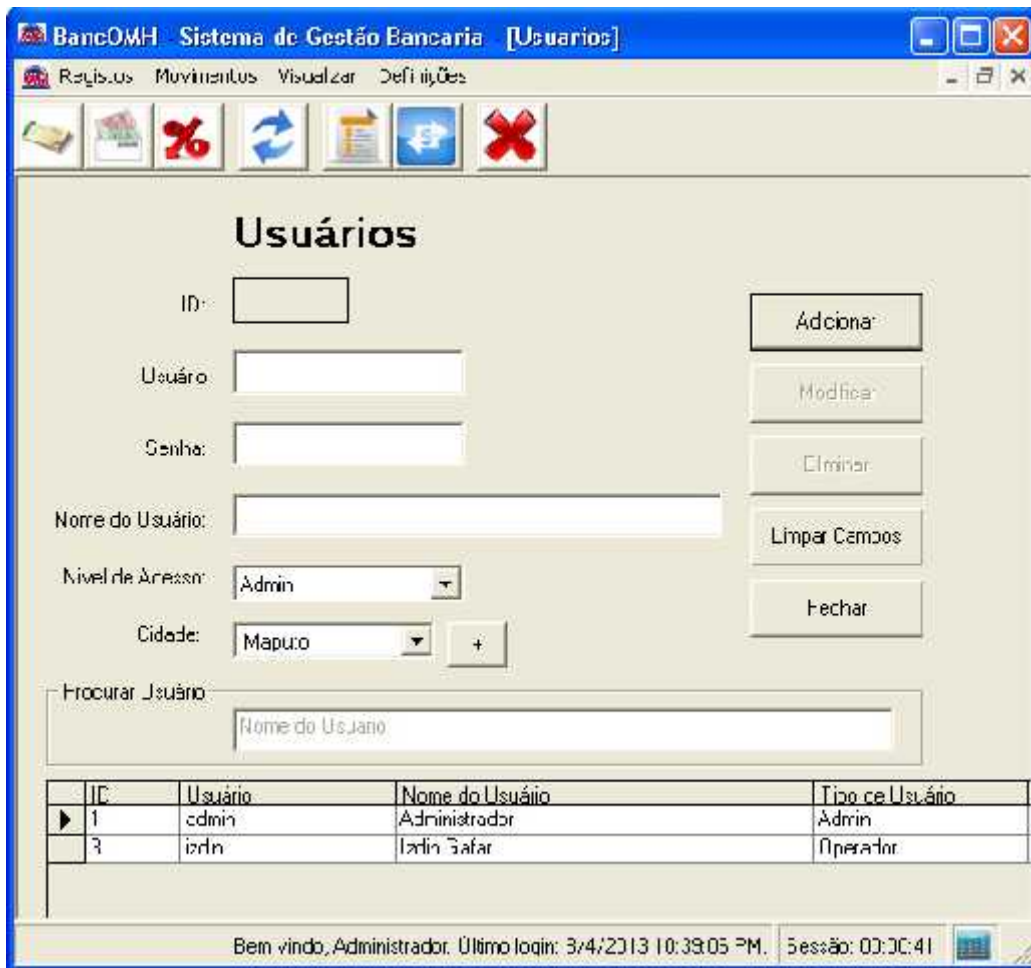


Figura 13– Tela dos Usuários



Figura 14– Tela de adição de cidades

5.2.3. Gestão de Moedas

Esta tela só pode ser acessada pelos administradores e operadores do sistema. Nesta tela o usuário faz as operações de criação, alteração e eliminação de moedas conforme ilustra a Figura 17. A leitura é automática acontecendo logo que a tela é carregada ou a partir do momento que aconteça alguma alteração no sistema relacionada com as moedas. São verificados os dados para evitar que se introduza a mesma moeda varias vezes.

| ID | Descrição | Sigla |
|----|--------------------|-------|
| 1 | Dolares Americanos | USD |
| 2 | Rands | ZAR |
| 3 | Metical | MZN |

Figura 15– Tela das Moedas

5.2.4. Gestão de Bancos

Nesta tela são feitas as operações de criação, edição, eliminação e alteração dos bancos. É uma tela simples e intuitiva. Mais uma vez são usados aqui os procedimentos de validação dos dados que verificam se o banco não foi previamente introduzido. Quando o usuário quer eliminar um banco, o sistema verifica se este banco está associado a alguma conta, evitando assim que se apague por engano.



Figura 16– Tela dos Bancos

5.2.5. Gestão de Contas

Aqui são efectuadas as operações relacionadas com as contas. Esta tela pode ser acedida pelos administradores e operadores, mas para os últimos, estes só podem criar contas para as cidades que eles estão associados.

Todas as contas devem estar ligadas a uma cidade, moeda e banco e por isso foram criados *combos* para forçar o usuário a introduzir dados validos. Caso o usuário deseje uma opção que não esteja no *combo* então o usuário pode adiciona-la em momento de execução. Estes botões permitem a criação e selecção rápida de cidades, bancos e moedas para evitar que o usuário tenha que sair da tela e ir a tela específica do outro módulo.

Os dados em **negrito** são verificados, e para o caso do saldo, é convertido para unidade monetária. Um algoritmo foi criado para tratamento especial deste campo. Operações como separação de três dígitos, introdução de ponto para separação das décimas e arredondamento do mesmo, caso seja necessário.

Quando a conta é eliminada, são eliminados também os movimentos associados a esta. Devido ao risco que ela exhibe esta operação de eliminação de contas só esta disponível

para os administradores, porém aparece para os operários mas vem desabilitada (Ver Figura 19).



Figura 17– Tela das Gestão das Contas

Quando criamos uma conta, um movimento de saldo inicial é adicionado a base de dados com ligação ao usuário que fez a operação.

5.3.MOVIMENTOS

Nestes módulos são feitas as operações relacionadas com os lançamentos dos movimentos diários. Este módulo só pode ser acessado pelos administradores e operadores, sendo que os últimos só têm acesso as contas da cidade onde estão registrados.

Por ser considerado o módulo mais usado do sistema, aqui tentou-se tirar o máximo proveito da acessibilidade, uso da tecla *tab* e das setas de cima e de baixo para poder

mudar de campos e guardar os movimentos. Todas as telas deste módulo contêm *links* para as outras telas de operações do mesmo módulo.

Todas as operações são lançadas na mesma tabela na base de dados. Os descritivos dos lançamentos variam de acordo com o tipo de operação para permitir que quando necessário se filtrem os cheques passados ou depósitos. Os saldos da conta são alterados de acordo.

5.3.1. Cheques

Nesta tela são feitos os lançamentos dos cheques passados. O usuário selecciona a conta que deseja fazer o lançamento, introduz a data do documento, o número do cheque, o beneficiário, o valor e o histórico, ou seja o descritivo da operação. Para o usuário com o nível de operador, só aparecerão listadas as contas de sua cidade no *combo* das contas. A tela é carregada contendo a data actual.

O usuário pode mudar de campos usando a tecla de cima e baixo ou também a tecla *tab*. Quando o usuário esta no último campo, histórico, ele pode gravar usando a tecla *enter*, o botão guardar ou a seta de baixo que chama o procedimento de gravação. Todos os campos são verificados, com excepção do campo histórico. Para o valor, os dados são tratados de forma especial, e convertidos para unidade monetária. Quando é feita a gravação, o próximo lançamento é por *default* para a mesma conta a menos que se especifique o contrário. Isto se deve ao fato de facilitar o usuário para não ter que seleccionar de novo a mesma conta no *combo*. O foco é então enviado para o campo da data do documento. A data é também inicializada com a ultima data usada.

Cheques

Conta: 80802-0.001 Barco: BCI
Descriçã: Conta 2 Maputo

Data: 05/08/2013

Ni. Cheque:

Num.:

Saldo actual: 15,750 49 MZN

Valor.: MZN

Historico:

[Depositos](#)
[Outros](#)
[Actualizar](#)

Guardar Limpar Campos Fechar

Figura 18– Tela dos Cheques Passados

5.3.2. Depósitos

Nesta tela são feitos os lançamentos dos depósitos. Esta tela funciona de forma semelhante a tela dos cheques. Nesta tela todos os campos são obrigatórios. Todos os depósitos efectuados têm um número em sequência, numero este que será preenchido no campo documento. Os proprietários solicitaram que se diferenciasse os depósitos em cheque dos depósitos em numerário. O histórico guardado na base de dados é formatado de acordo com o tipo de operação, para facilitar na leitura dos extractos.

Depositos

Conta: 69452.10.001 Banco: MozaBanco
Descrição: Conta 1 Maputo

Data: 05/08/2013

Documento:

Saldo actual: 14,908.71 USD

Valor: USD

Cheque Numerário

Historico:

[Cheques](#)
[Outros](#)
[Actualizar](#)

Guardar Limpar Campos Fechar

Figura 19– Tela dos Depósitos

5.3.3. Tarifas Bancárias e Outros

Nesta tela são feitas todas as operações diferentes de cheques e depósitos. Estes podem ser juros, comissões bancárias, etc. Todos os campos são obrigatórios e o usuário deve especificar que tipo de movimento se trata, se é um débito ou crédito. Mensagens apropriadas são adicionadas ao campo apropriado da base de dados para permitirem a filtragem dos dados quando necessário.

Tarifas Bancárias e Outros

Conta: 69452 10 001 Banco: MozaBanco
 Descrição: Conta 1 Maputo
 Saldo actual: 14,508.71 USD

Data: 05/08/2013

Documento:

Débito Crédito

Valor: USD

Historico:

[Cheques](#)
[Depositos](#)
[Actualizar](#)

Guardar Limpar Campos Fechar

Figura 20– Tela de Outros movimentos

5.3.4. Actualizar Lançamentos

As telas anteriores são somente para operações de adição de movimentos. Nesta tela fazemos a leitura, alteração e eliminação dos lançamentos. O usuário selecciona a conta que deseja verificar, e uma tabela com os lançamentos é mostrada. Estes lançamentos podem ser filtrados por data, histórico, valor e saldo remanescente após a operação. Isto permite uma flexibilidade na pesquisa quando tratamos de uma conta com elevado número de lançamentos.

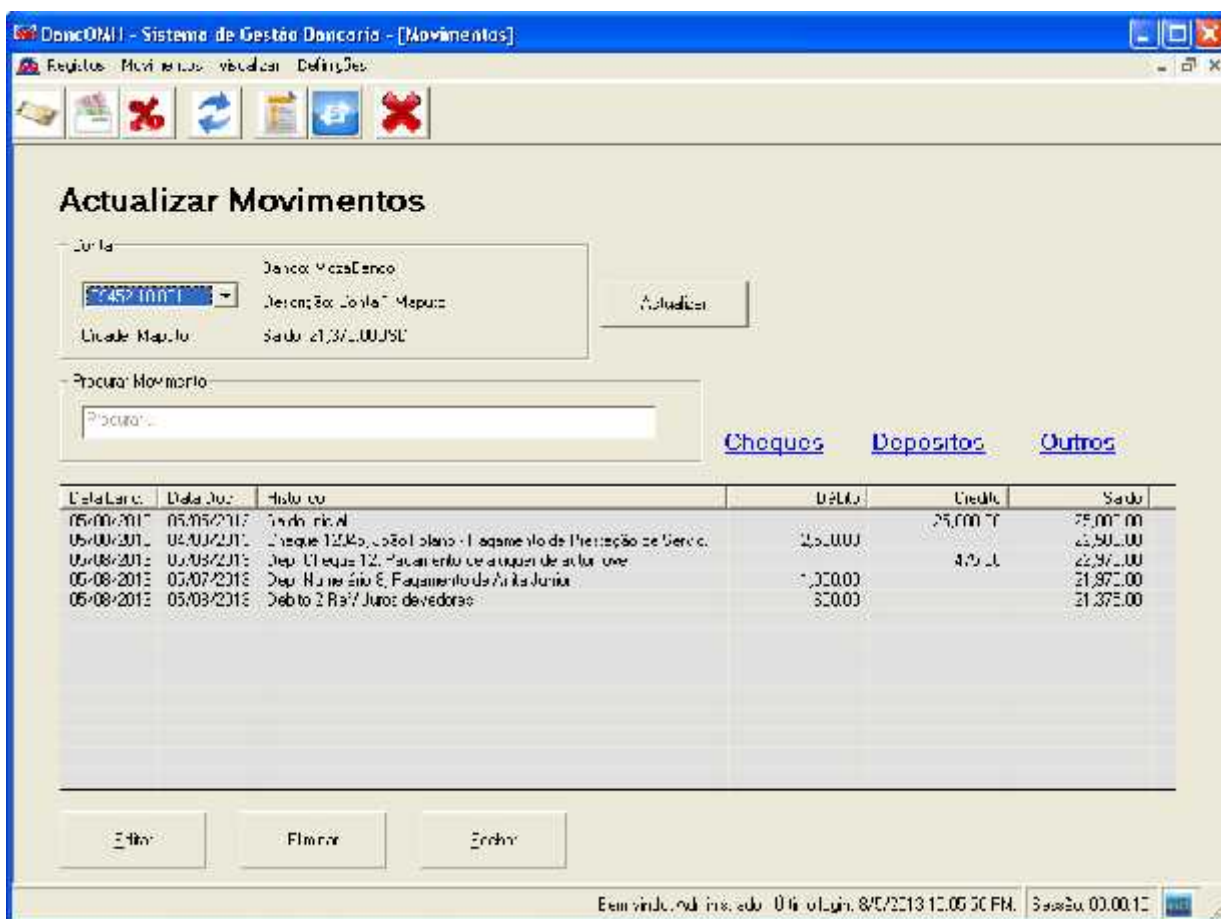


Figura 21– Tela das Actualizações de Movimentos

A lista dos movimentos é actualizada a partir do momento que é editado o texto de filtragem para poder mostrar os movimentos que se enquadrem na busca. A cada carácter introduzido a lista é actualizada. A seguir são mostradas as operações que podem ser feitas nesta tela.

5.3.4.1. Eliminar

Para eliminar um movimento o usuário selecciona o lançamento e clica no botão eliminar. O usuário é então confrontado com uma caixa de diálogo perguntando se deseja continuar. Caso a resposta seja sim, então um procedimento de eliminação tem o seu início. Neste procedimento acontecem múltiplas ligações a base de dados. Primeiro elimina-se o lançamento, depois são alterados todos os movimentos consequentes ao movimento em questão, e finalmente, altera-se o saldo da conta para poder reflectir o novo valor.

5.3.4.2. Modificar

Para editar os movimentos o usuário deve seleccionar na lista o movimento que deseja alterar e fazer duplo clique ou então clicar no botão editar. Uma tela aparece mostrando os dados do lançamento conforme mostra a Figura 24. O usuário faz as alterações que deseja, muda o tipo de lançamento de débito para crédito ou vice-versa caso seja necessário. Todos os campos aqui são verificados. Em relação ao histórico, cabe ao usuário criar uma mensagem descritiva. Nesta tela também temos um botão que permite fazer a eliminação do lançamento. Transacções de alteração de saldos noutros movimentos são realizadas caso seja necessário.



A imagem mostra uma janela de software intitulada "Movimento". O formulário contém os seguintes campos e controles:

- ID:** Campo de texto com o valor "114".
- Data Doc.:** Campo de data com o valor "05/03/2013".
- Histórico:** Campo de texto com o valor "Crédito 3 Ref/Juros Credores - Bónus".
- Valor:** Campo de texto com o valor "200.00" e a unidade "USD".
- Saldo:** Campo de texto com o valor "14.508,71" e a unidade "USD".

À direita do formulário, há três botões: "Eliminar", "Guardar" e "Cancelar".

Figura 22– Tela de Edição de um movimento

5.4.VISUALIZAR

Neste módulo estão presentes telas para a pesquisa de movimentos e também para a impressão dos mesmos se necessário. Todos os relatórios gerados neste módulo foram criados usando o *DataReport*, uma ferramenta de criação de relatórios incluída no Visual Basic 6.

Como foi explicado anteriormente, este é apenas um módulo de leitura. Sendo assim, todos usuários têm acesso ao mesmo. Os relatórios gerados neste módulo estão ilustrados na secção dos Anexos deste trabalho.

5.4.1. Cheques Passados

Nesta tela o usuário consulta os cheques passados durante um certo período de tempo. Primeiro o usuário selecciona uma cidade, e em seguida a conta, depois o período. Uma tabela com os cheques passados neste período é mostrada. Caso o usuário deseje imprimir então ele clica no botão imprimir.



| # | Data Doc. | Descrição | Valor |
|---|------------|--|----------|
| 1 | 03-08-2013 | Cheque 67, TDM - Telefone Fixo | 340.00 |
| 2 | 04-08-2013 | Cheque 12345, João Folano - Pagamento de Prestação de Serviços | 2,500.00 |
| 3 | 04-08-2013 | Cheque 3424, TDM - Fax | 65.00 |
| 4 | 06-08-2013 | Cheque 3454, EDM - Energia | 2,534.00 |

Figura 23– Pesquisa de Cheques Passados

5.4.2. Depósitos

Neste módulo, o usuário tem acesso aos depósitos efectuados numa conta durante um certo período. Criamos duas vertentes para este módulo, depósitos em cheque e depósitos em numerário. Os dois funcionam de forma semelhante, apenas o resultado com resultados

diferentes. Todos os lançamentos têm um campo reservado na base de dados que permite a discriminação dos mesmos.

The screenshot shows a window titled "Depósitos em Cheque". It contains the following fields and controls:

- Cidade:** Maputo (dropdown)
- Banco:** MozaBanco
- Conta:** 69452.10.001 (dropdown)
- Descrição:** Conta 1 Maputo
- Moeda:** USD
- Periodo:** 10/07/2013 a 09/08/2013 (date pickers)
- Imprimir:** button

| # | Data Doc. | Descrição | Valor |
|---|------------|---------------------------------------|--------|
| 1 | 02-08-2013 | Dep. Cheque 34, Merec Industries | 244.00 |
| 2 | 06-08-2013 | Dep. Cheque 23, Africom - Factura 123 | 222.00 |

Figura 24– Pesquisa de Depósitos em Cheque

5.4.3. Extractos

Neste módulo, o usuário tem acesso ao extracto da conta. O usuário deve seleccionar uma conta e um período de tempo do extracto.

The screenshot shows a window titled "Extractos". It contains the following fields and controls:

- Cidade:** Maputo (dropdown)
- Banco:** MozaBanco
- Conta:** 69452.10.001 (dropdown)
- Descrição:** Conta 1 Maputo
- Moeda:** USD
- Periodo:** 07/07/2013 a 06/08/2013 (date pickers)
- Imprimir:** button

Figura 25– Pesquisa de Extracto de Conta

5.4.4. Saldos

Nesta tela o usuário verifica o saldo da conta. Na sequência do pedido dos usuários do sistema, eles pediram que se pudesse verificar não só o saldo actual, mas também o saldo numa data anterior. Um algoritmo apropriado foi criado com este fim. Aqui o saldo é mostrado para todas as contas da cidade seleccionada, e na data seleccionada.



Figura 26– Pesquisa de Saldos para cidade seleccionada

5.5.DEFINIÇÕES

Esta é uma tela simples que contem informações sobre o sistema, a versão e o desenvolvedor do sistema.



Figura 27– Tela “Sobre”

5.6.INSTALAÇÃO DO SISTEMA

O bancOMH tem um *setup* que faz a instalação personalizada do sistema. É um processo simples. O usuário só tem que fazer duplo clique no ficheiro de *setup* seguir as instruções do instalador. A seguir são descritos os requisitos do sistema, quer para o cliente quer para o servidor.

5.6.1. Requisitos para o Cliente

O bancOMH é um aplicativo Desktop, portanto, do ponto de vista do cliente, é necessário que o sistema seja instalado num computador com acesso ao servidor de base de dados. O sistema corre nos mais populares sistemas operativos Windows, utilizados pelos usuários finais da aplicação.

5.6.2. Requisitos para o Servidor

Para a implementação, é necessário, um servidor de base de dados que pode ser implementado tanto em plataformas Linux quanto em Windows. A base de dados utilizada foi a MySQL, porem nada impede que os dados e as tabelas sejam exportados para outras bases de dados.

CAPITULO 6 - CONCLUSÕES E RECOMENDAÇÕES

A forma inadequada de armazenamento da informação referente ao controlo interno das contas bancárias da empresa Organizações Mamade Hanif gerava uma dificuldade de obtenção informação decisiva no momento que esta se fazia necessária.

O bancOMH vem para oferecer a empresa, transacções que possibilitam o armazenamento das informações e relatórios que, em qualquer momento, podem ser acedidos e analisados, tudo isso de forma transparente e imediata. Possibilita ainda o acesso do sistema por vários usuários, entre eles administradores, operadores e proprietários da empresa, todos eles controlados por um sistema de perfil de permissões, dando maior agilidade e segurança a execução das tarefas. O sistema possui também interface limpa e simplificada, caixas de selecção de dados para facilitar na entrada das informações, tudo isso faz com que a interface seja mais amigável e confiável para quem esteja acedendo o bancOMH.

Levando-se em conta a antiga situação em que se encontrava a empresa e a forma por eles utilizada para armazenar as informações pertinentes a manutenção do seu negócio, conclui-se que o bancOMH, de forma simples, atende as principais necessidades da empresa eficazmente. Verifica-se um aumento do fluxo interno de informação e integração entre os departamentos distintos na empresa, o rendimento dos funcionários vem melhorando consideravelmente. Aumentou a precisão na obtenção de informações e relatórios, já que os dados passam a ser criteriosamente armazenados. Os proprietários sentem-se mais confiantes sabendo que podem tomar decisões baseando-se em dados reais e acima de tudo as informações estão guardadas em lugares seguros.

6.1.RECOMENDAÇÕES

Este trabalho serviu mais uma vez para mostrar que é preciso ter um sistema destes dentro das empresas, não se trata de um luxo mas sim de uma necessidade. Os usuários são geralmente resistentes à mudança mas é sempre bom fazê-los entender que os benefícios da TI nas empresas estão ligados directamente à vantagem competitiva. Dentre esses benefícios, inclui-se suporte à tomada de decisão, controle de produção e qualidade, gestão da informação, entre vários outros, oferecendo mais facilidade de administração e gestão interna.

Assim, ao se obter melhor gestão da informação interna da empresa, esse fluxo de informações harmónico se reflectirá também no ambiente externo, proporcionando diferencial competitivo no mercado de trabalho actual.

6.2.TRABALHOS FUTUROS

A tecnologia vai progredindo e ultimamente tem-se verificado uma tendência de apostar na mobilidade. Para o futuro pode se fazer uma interface Web com suporte a dispositivos móveis para permitir que os proprietários tenham acesso as informações a qualquer hora e lugar desde que tenham um computador, celular ou *tablet* com acesso a internet.

ANEXO A – RELATÓRIO DE CHEQUES PASSADOS



Av. Patrice Lumumba N 403 R/C
Tel-21320685 - Fax-21321819
Email: organizacoesmh.mp@gmail.com

Data: 13-08-2013

Conta 1 Maputo
Nº. de Conta: 69452.10.001

Moeda: MZN
Banco: MozaBanco

Relação de Cheques Passados

Período: 12-07-2013 a 13-08-2013

Pag. 1/1

| <u>Data Doc.</u> | <u>Historico</u> | <u>Valor</u> |
|------------------|--|-----------------|
| 12-08-2013 | Cheque 2346, João Antoninho - Prestação de Serviços | 3,500.00 |
| 12-08-2013 | Cheque 2348, Dino Martin - Empréstimo | 45.20 |
| 13-08-2013 | Cheque 2349, Ferragens Maputo - Compra de Material Diverso | 5,601.23 |
| | Total | 9,146.43 |

ANEXO B – RELATÓRIO DE DEPÓSITOS EM CHEQUE



Av. Patrice Lumumba N 403 R/C
Tel-21320685 - Fax-21321819
Email: organizacoesmh.mp@gmail.com

Data: 13-08-2013

Conta 1 Maputo

Nº. de Conta: 69452.10.001

Moeda: MZN

Banco: MozaBanco

Relação de Depósitos em Cheque

Período: 12-05-2013 a 15-08-2013

Pag. 1/1

| <u>Data Doc.</u> | <u>Historico</u> | <u>Valor</u> |
|------------------|---|------------------|
| 09-08-2013 | Dep. Cheque 1, ElectroMetal - Factura 34/13 | 5,000.00 |
| 10-08-2013 | Dep. Cheque 3, EDM - Compra de 10 rolos cabos 2x10 | 25,000.00 |
| 13-08-2013 | Dep. Cheque 7, Merez Industries - Factura 34/12 | 45.46 |
| 13-08-2013 | Dep. Cheque 8, Empresa X - Compra de Material diverso | 3,400.00 |
| | Total | 33,445.46 |

ANEXO C – RELATÓRIO DE SALDOS



Av. Patrice Lumumba N 403 R/C
Tel-21320685 - Fax-21321819
Email: organizacoeshm.mp@gmail.com

Data: 13-08-2013

Saldos Bancários - Maputo

| Descrição | Nº. de Conta | Saldo | Moeda |
|----------------|---------------|-----------|-------|
| Conta 1 Maputo | 69452.10.001 | 25,000.00 | USD |
| Conta 2 Maputo | 80802.10.001 | 10,000.00 | MZN |
| Conta 3 Maputo | 126573.10.002 | 5,800.00 | USD |
| Conta 4 Maputo | 78313.10.002 | 23,457.89 | USD |

ANEXO D – RELATÓRIO DE EXTRACTOS



Av. Patrice Lumumba N 403 R/C
Tel-21320685 - Fax-21321819
Email: organizacoesmh.mp@gmail.com

Data: 13-08-2013

Conta 1 Maputo
Nº. de Conta: 69452.10.001

Moeda: MZN
Banco: MozaBanco

Extracto da Conta

Período: 14-07-2013 a 13-08-2013

| Data | Tipo de Movimento | Debito | Credito | Saldo |
|------------|--|-----------|--------------|------------------|
| | Pag. 1/1 | | | |
| 13-08-2013 | Saldo Inicial | 0.00 | 25.000.00 | 25,000.00 |
| 13-08-2013 | Cheque 2345, EDM - Pagamento de Energia | 3,500.00 | 0.00 | 21,500.00 |
| 13-08-2013 | Cheque 2346, Sanal, Lda. - Compra de 600 Sacos | 2,500.00 | 0.00 | 19,000.00 |
| 13-08-2013 | Dep. Cheque 1, ElectroMetal - Factura 34/13 | 0.00 | 5,000.00 | 24,000.00 |
| 13-08-2013 | Dep. Numerário 2, ElectroMetal - Diferença de Valor da factura | 0.00 | 25.67 | 24,025.67 |
| 13-08-2013 | Dep. Cheque 3, EDM - Compra de 10 rolos cabos 2x10 | 0.00 | 25,000.00 | 49,025.67 |
| 13-08-2013 | Crédito 123 Ref/ Juros Credores | 0.00 | 43.64 | 49,069.31 |
| 13-08-2013 | Débito 56 Ref/ Crédito de Renda - Mes de Setembro | 20,001.20 | 0.00 | 29,068.11 |
| 13-08-2013 | Cheque 2346, João Antoninho - Prestação de Serviços | 3,500.00 | 0.00 | 25,568.11 |
| 13-08-2013 | Cheque 2348, Dino Martin - Empréstimo | 45.20 | 0.00 | 25,522.91 |
| 13-08-2013 | Dep. Cheque 7, Merc Industries - Factura 34/12 | 0.00 | 45.46 | 25,568.37 |
| 13-08-2013 | Dep. Cheque 8, Empresa X - Compra de Material diverso | 0.00 | 3,400.00 | 28,968.37 |
| 13-08-2013 | Cheque 2349, Ferragens Maputo - Compra de Material Diverso | 5,601.23 | 0.00 | 23,367.14 |
| | | | Total | 23,367.14 |

ANEXO E – PLANO DE ENTREVISTA

Será feita uma entrevista aberta, onde haverá lugar a uma discussão sobre o sistema com os futuros usuários deste. Serão abordados tópicos como:

- O fluxo das suas actividades diárias.
- Que tipos de dados são introduzidos neste processo.
- Quais são os resultados esperados destes dados
- O que os usuários acham deste sistema actual, quais são as suas fraquezas
- Como podemos melhorar o sistema
- O que o novo sistema deve ser capaz de fazer
- Quantas pessoas vão usar o sistema
- Quem deve ter acesso ao sistema
- Em que sistema operativo deve correr
- Que tipo de relatórios se espera que o sistema deva produzir

BIBLIOGRAFIA

Albertin, A. L. (2002). *Administração de informática: funções e factores críticos de sucesso*. São Paulo: Atlas.

Da Silva, N. P. (2007). *Análise e estruturas de sistemas de informação*. São Paulo: Érica.

Fowler, M. (1999). *UML Distilled Second Edition A Brief Guide to the Standard Object Modeling Language*. Boston: Addison Wesley.

Galliers, R. (1987). *Information analysis: Selected readings*. Boston: Addison Wesley.

Mañas, A. V. (1999). *Administração de sistemas de informação*. São Paulo: Érica.

Mcgee, J. & Prusak, L. (1994). *Gerenciamento estratégico da informação: aumente a competitividade e a eficiência de sua empresa utilizando a informação como uma ferramenta estratégica*. Rio de Janeiro: Campus.

Sêmola, M. (2003). *Gestão da segurança da informação: uma visão executiva*. Rio de Janeiro: Campus.

Silva, A. M. & Videira, C. (2001). *UML, Metodologias e Ferramentas CASE*. Porto: Centro Atlântico.

Slack, N., Chambers, S. & Johnston, R. (2002). *Administração da produção*. São Paulo: Atlas.