



**UNIVERSIDADE EDUARDO MONDLANE**

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

TEMA:

**Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM**

CASO DE ESTUDO: DSS

**Autor:** Celso Meirelles Rodolfo Adamo



**UNIVERSIDADE EDUARDO MONDLANE**

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

TEMA:

**Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM**

CASO DE ESTUDO: DSS

**Autor:** Celso Meirelles Rodolfo Adamo

**Supervisora:** dra. Rossana Carimo Soares

**Co-Supervisor:** dr. Carlos da Silva

**MAPUTO, MARÇO DE 2011**

## **Dedicatória**

Aos meus pais, **Francisco Álvaro Marta Adamo e Ângela Maria Lucas Rodolfo Adamo.**

Aos meus queridos irmãos, **Edvaldo Marcio Rodolfo Adamo, Francisco Álvaro Rodolfo Adamo Jr. e Alexandre Marta Rodolfo Adamo.**

## Agradecimentos

Ao longo do meu curso e durante o período de elaboração do trabalho de final de curso, houve contributos de várias pessoas, dentre estas professores, familiares, amigos e colegas. Sendo assim, agradeço de forma especial as seguintes pessoas:

Primeiramente, agradeço aos meus pais, pelo carinho, força e ensinamentos que me foram proporcionados.

A minha prima Tânia pela correcção de erros ortográficos e sugestões por ela proporcionadas.

A todos funcionários da Direcção dos Serviços Sociais (DSS) pela colaboração, disponibilização de dados que foram fundamentais para realização desta monografia e pelo tempo que me foi concebido para realização das entrevistas, especialmente ao dr. Macedo, dr. Cossa e sr. Evaristo.

Aos meus colegas e amigos que directa ou indirectamente deram o seu contributo para concretização deste trabalho, especialmente ao Elísio Leonardo e Raimundo Chongo pelos conhecimentos partilhados e pela convivência agradável durante o curso.

Por fim, agradeço de forma muito especial aos meus supervisores, dra. Rossana Carimo Soares (Supervisora) e dr. Carlos da Silva (Co-Supervisor) pela disponibilidade, motivação e orientação que me foi concebida durante a elaboração desta dissertação.

## **Declaração de Honra**

Declaro por minha honra que o presente trabalho foi resultado da minha própria investigação e o mesmo foi concebido para ser submetido como trabalho para obtenção do grau de Licenciatura em Informática na Faculdade de Ciências da Universidade Eduardo Mondlane, Departamento de Matemática e Informática.

Maputo, Março de 2011

O Autor

---

(Celso Meirelles Rodolfo Adamo)

## Resumo

O objecto de pesquisa do presente trabalho é a Direcção dos Serviços Sociais (DSS), um órgão da Universidade Eduardo Mondlane (UEM) que presta serviços de apoio aos estudantes com objectivo de proporcionar-lhes as melhores condições de estudo. O volume de dados e informação referente ao processo de alojamento e pagamento de mensalidades da DSS cresce dia após dia, o que torna a organização e processo de tomada de decisão mais complexos.

Com intenção de se fazer uma gestão eficaz do recurso informação, que é essencial para tomada de decisão em qualquer organização, propôs-se desenvolver um modelo que permite efectuar a gestão dos alojamentos nas residências universitárias da UEM usando tecnologias *Web*, tendo como condição necessária o acesso à *Internet*.

Para se alcançar o objectivo citado acima, que coincide com a proposta de solução, fez-se ao longo do trabalho, a comparação de algumas metodologias de desenvolvimento de *software*, entrevistou-se alguns funcionários da DSS, consultou-se e analisou-se documentos referentes à DSS, e finalmente desenhou-se o modelo julgado ideal para minimizar ou acabar com os problemas encarados pela DSS e pelos estudantes residentes nas residências universitárias da UEM.

Entretanto, com a realização desta dissertação, espera-se como resultado um modelo que permita resolver problemas tais como: a demora na emissão dos recibos comprovativos de pagamento de mensalidades e a dificuldade de obter informação actualizada sobre os estudantes.

## Sumário

Dedicatória .....	I
Agradecimentos .....	II
Declaração de Honra .....	III
Resumo .....	IV
Glossário .....	XI
Capítulo I – Introdução .....	1
1- Contextualização .....	1
2- Definição do Problema .....	2
3- Objectivos .....	4
3.1- Geral.....	4
3.2- Específicos.....	4
4- Estrutura do Trabalho .....	5
Capítulo II – Metodologia .....	6
Capítulo III – Revisão Bibliográfica.....	10
1- Introdução.....	10
2- Modelo de desenvolvimento Incremental .....	11
2.1- Introdução.....	11
2.2- Características.....	12
2.3- Vantagens e desvantagens .....	12
3- Processo de Modelação Unificado.....	13
3.1- Introdução.....	13

3.2- Características.....	14
3.3- Actividades .....	15
3.4- Fases.....	16
3.5- Vantagens e desvantagens .....	18
4- Programação Extrema .....	18
4.1- Introdução.....	18
4.2- Valores .....	19
4.3- Princípios ou Práticas do XP .....	20
4.4- Actividades .....	22
4.5- Vantagens e desvantagens .....	23
Capítulo IV – Caso de Estudo: Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM.....	25
1- Descrição da Situação Actual.....	25
2- Residências Universitárias .....	26
3- Processos de Negócio .....	29
3.1- Processo de Alojamento e Desalojamento.....	29
3.2- Processo de permuta e troca de quartos.....	30
4- Organograma .....	31
Capítulo V – Modelo Proposto de Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM.....	32
1- Descrição do Modelo Proposto .....	32
2- Arquitectura do Modelo Proposto .....	34
3- Funcionamento do Modelo Proposto.....	35
4- Modelação do Modelo Proposto.....	37



4.1- Descrição dos Actores .....	37
4.2- Casos de Uso .....	38
4.3- Diagramas de Casos de Uso .....	40
4.4- Descrição Estruturada dos Casos de Uso .....	42
4.5- Diagrama de Classes .....	50
4.6- Diagrama de Actividades .....	51
4.7- Diagrama de Sequência .....	53
4.7- Diagrama de Estados .....	55
Capítulo VI – Conclusões e Recomendações .....	56
1- Conclusões .....	56
2- Recomendações .....	56
Bibliografia.....	58
Referenciada .....	58
Não Referenciada.....	59
Anexos.....	60
Anexo 1: Guião de Entrevista.....	60
Anexo 2: Ficha de Internamento.....	62
Anexo 3: Contrato de arrendamento de cama .....	63
Anexo 4: Guia de Autorização de Ocupação de Cama .....	67
Anexo 5: Manual do Utilizador .....	68

## **Índice de Tabelas**

Tabela 1 – Lista de Endereços das Residências Universitárias .....	28
--	----

## Índice de Figuras

Figura 1- Relação entre as fases e actividades do PMU .....	14
Figura 2 – Organograma da DSS.....	31
Figura 3 - Fluxo de informação do modelo proposto .....	34
Figura 4 - Arquitectura do modelo proposto .....	35
Figura 5 – Diagrama de Casos de Uso para actores emissores (1/2).....	40
Figura 6 – Diagrama de Casos de Uso para actores receptores (2/2) .....	41
Figura 7 – Diagrama de Classes .....	50
Figura 8 – Diagrama de Actividades: Registrar Pagamentos .....	51
Figura 9 – Diagrama de Actividades: Registrar Estudante.....	52
Figura 10 – Diagrama de Sequência: Registrar Estudante .....	53
Figura 11 – Diagrama de Sequência: Registrar Pagamento .....	53
Figura 12 – Diagrama de Sequência: Registrar Usuário .....	54
Figura 13 – Diagrama de Sequência: Alojamento do estudante .....	54
Figura 14 – Diagrama de Estado: Usuário .....	55
Figura 15 – Diagrama de Estado: Pagamento .....	55

## Lista de Abreviaturas

BA – Bolsa de Alojamento

BC – Bolsa Completa

BD – Base de Dados

BI – Bilhete de Identidade

BR – Bolsa Reduzida

DA – Departamento de Alojamento

DAF – Departamento de Administração e Finanças

DMI – Departamento de Matemática e Informática

DRA – Direcção do Registo Académico

DSS – Direcção dos Serviços Sociais

PMU – Processo de Modelação Unificado

UEM – Universidade Eduardo Mondlane

RU – Residência Universitária

SGBD – Sistema de Gestão de Base de Dados

SI – Sistemas de Informação

TI – Tecnologias de Informação

XP – *Extreme Programming*

## Glossário

**Abordagem Orientada a Objectos** – é uma abordagem que permite ver o mundo como objectos com estrutura de dados e comportamentos.

**Aplicação Web** – é o termo utilizado para designar de forma geral, sistemas informáticos projectados para utilização através de navegador (em inglês *browser*), na *Internet* ou *Intranet* (redes privadas).

**Base de Dados (BD)** – é uma colecção de dados operacionais armazenados, usados pelos sistemas de aplicações de uma empresa específica.

**Código aberto (em inglês Open Source)** – é um termo criado pela Open Source Initiative (OSI) e refere-se a um *software* também conhecido por *software* livre. Um *software* livre pode ser usado, copiado, estudado e redistribuído sem restrições.

**Ferramentas CASE** – são sistemas de *software*, cujo objectivo é fornecer suporte automatizado para as actividades do processo de desenvolvimento de *software*, como a engenharia de requisitos, o projecto, o desenvolvimento do programa e os testes.

**Framework** – é um conjunto de classes reutilizáveis que compõem um projecto para classes específicas de *software*.

**Internet** – é uma rede global de computadores interligados, que se comunicam entre si através de redes de telecomunicações existentes.

**Java** – é uma linguagem de programação orientada a objectos, desenvolvida pela Sun Microsystems. É muito popular e crescentemente usada no mundo da *Web*.

**JavaScript** – é uma linguagem de Scripts (orientada a objectos) desenhada para *Web*. Esta permite que os desenvolvedores *Web* incluam um alto nível de interactividade em páginas *Web*.

**Navegador Web (em inglês Web Browser)** – é um programa que permite nos visualizar e explorar a informação na *Web*.

**Página Web** – é um documento na *Web* que pode incluir textos, fotos, som e vídeos.

**Plug-in** – é um programa de computador que permite adicionar funções a outros programas maiores, provendo alguma funcionalidade específica.

**Servidor Web** – é uma entidade composta por *hardware* ou *software* que fornece serviços de consulta por hipertexto (WWW) na *Internet* ou *Intranet*.

**Sistema de Gestão de Base de Dados (SGBD)** – é um conjunto de *software*, destinado a gerir todo o armazenamento e manipulação dos dados do sistema, fazendo a interface entre o nível aplicacional e a base de dados propriamente dita.

**Sistema de Informação (SI)** – é uma entidade sociotécnica que reúne, guarda, processa e faculta informação relevante para a organização (ou para a sociedade), de modo a torna-la acessível e útil para aqueles que desejam (e possam) utilizarem.

**Software** – são programas de computador e toda documentação associada, ou seja, é a componente lógica da informática (programas, aplicações, sistemas operativos) que comanda o *hardware*.

**Tecnologia de Informação (TI)** – é um conjunto de equipamentos e suportes lógicos (*hardware* e *software*) que permitem executar tarefas como aquisição, transmissão, armazenamento, recuperação e exposição de dados.

**URL (Uniform resource Locator)** – é o endereço único atribuído a cada página *Web*. Conhecendo URL podes mostrar instantaneamente qualquer página *Web*.

**Web** – é o termo conhecido por WWW.

**World Wide Web (WWW)** – rede virtual de informação, composta por um conjunto de máquinas denominadas servidores *Web*.

## Capítulo I – Introdução

### 1- Contextualização

A tomada de decisões nas organizações é um processo complexo, dada a quantidade de informação em jogo, a sua complexidade e a frequência com que esta informação se altera (PEREIRA, 1998). Segundo RODRIGUES (2002) as Tecnologias de Informação (TI) funcionam como meio de suporte e de melhoria das actividades das organizações e estas continuam a alterar profundamente o modo as organizações evoluem e os negócios se processam.

A Universidade Eduardo Mondlane (UEM) – é uma organização pública de ensino superior que tem como tarefa principal formar técnicos de nível superior, capazes de produzir, aplicar, difundir a cultura, a ciência e a técnica ao serviço do desenvolvimento do país e do mundo (CONSELHO UNIVERSITÁRIO DA UEM, 2010). Esta possui vários docentes, estudantes, funcionários que se encontram distribuídos pelas diversas faculdades, departamentos, direcções, etc.

Dentre as diversas direcções existentes, a escolhida para o objecto de pesquisa é a Direcção dos Serviços Sociais (DSS) da universidade, especificamente o Departamento de Alojamento localizado no campus universitário da mesma, na cidade do Maputo.

Actualmente, a DSS utiliza um sistema manual que permite efectuar a gestão do processo de alojamento e pagamento de mensalidades para arrendamento de camas e neste constatou-se alguns problemas relacionados a eficiência e gestão dos mesmos.

Com intuito de resolver problemas tais como: a dificuldade de obter informação actualizada sobre estudantes, a dificuldade de elaborar estatísticas sobre as residências universitárias, e a demora na emissão de recibos comprovativos de pagamentos de mensalidades de arrendamento de camas, propõe-se, efectuar este trabalho, cujo objectivo é desenvolver um modelo de sistema de gestão de alojamentos para residências universitárias da UEM com recurso a tecnologia *Web*.

Na metodologia desta dissertação são abordadas de forma implícita: a estratégia de pesquisa, as técnicas de colheita de dados, as técnicas de análise de dados, o tipo de abordagem, as técnicas de modelação, a linguagem de programação, o Sistema de Gestão de Base de Dados (SGBD) e todas as ferramentas de apoio usadas durante o desenvolvimento do Sistema de Informação (SI), pois estes aspectos foram essenciais para se alcançar os objectivos específicos que são cruciais para se concretizar o objectivo geral.

## **2- Definição do Problema**

O Departamento de Alojamento (DA) faz parte da DSS e é responsável pelo alojamento dos estudantes. Sendo assim, no início de cada ano lectivo, a DSS recebe uma lista com uma certa quantidade de estudantes bolseiros enviada pela Direcção do Registo Académico (DRA). Neste mesmo período, a DSS também recebe cartas de pedidos de arrendamento de camas, dos estudantes não bolseiros e estudantes que perderam a Bolsa Completa (BC).

A quantidade de estudantes enviada pela DRA pode ser superior a capacidade de alojamento disponível pela DSS, pois, não se sabe exactamente quantas vagas existem em cada residência. Estas dificuldades de elaboração de estatísticas sobre as residências universitárias surgem pelo facto do processo de tomada de decisões na organização ser complexo, devido a quantidade de informação referente aos estudantes (processos estudantis) armazenados na DSS, a sua complexidade e a frequência com que esta informação se altera.

Ao longo dum certo período (tempo de duração do curso), o estudante pode perder a BC passando a usufruir apenas da Bolsa Reduzida (BR), perdendo assim a regalia de alojamento. Perdendo esta regalia, se o estudante quiser permanecer na residência universitária, deve depositar mensalmente um determinado valor referente ao arrendamento de cama na conta da DSS no banco Standard Bank. Após o depósito no banco, lhe é retornado um recibo comprovativo de depósito do banco.



Este recibo pode ser entregue ao administrador da residência universitária onde o estudante habita ou ao DA. Se entregue ao administrador, este no fim do mês, deve encaminhar todos os recibos comprovativos de depósito dos estudantes ao DA.

Por sua vez, o DA encarrega-se de entregar os novos processos de estudantes e os recibos comprovativos de depósito, respectivamente, a residência onde o estudante reside (ou irá residir) e a DSS, concretamente ao Departamento de Administração e Finanças (DAF). Depois de se verificar que os recibos comprovativos de depósito são válidos, a DAF retorna recibos comprovativos de pagamento de mensalidades de arrendamento de cama aos administradores.

Dado o elevado número da população estudantil que se envolve neste processo, este pode durar meses, provocando deste modo, uma demora na emissão de recibos comprovativos de pagamentos de mensalidades de arrendamento de camas, incorrendo assim o estudante ao risco de em algum momento ser considerado em situação ilegal por falta de pagamento da mensalidade. Caso o estudante seja considerado ilegal, este é expulso, perde direito sobre a cama arrendada e não pode mais efectuar a renovação do contrato referente ao arrendamento de cama com à DSS.

Salientar que, pelo facto dos registos de novos estudantes realizarem-se no DA e os estudantes poderem mudar de quartos ou residências apresentando uma justificação aceitável, os administradores não tem acesso a esta informação em tempo real, isto é, há dificuldades de obter informação actualizada sobre os estudantes.

Para resolver os problemas citados neste capítulo, propõe-se, desenvolver uma aplicação *Web* de gestão de alojamentos nas residências universitárias da UEM, pois, com a implementação da mesma, a informação ficará disponível para todos os níveis de gestão que precisam desta para tomar decisões em tempo útil, haverá facilidades de elaborar estatísticas (a aplicação disponibilizará diversos tipos de relatórios sobre a situação das residências universitárias) e deste modo permitirá uma gestão melhorada dos recursos alocados a população estudantil bolseira da UEM.

### **3- Objectivos**

Os objectivos desta dissertação são:

#### **3.1- Geral**

- ✚ Desenvolver um modelo de sistema de gestão de alojamentos para as residências universitárias da UEM com recurso a tecnologias *Web*.

#### **3.2- Específicos**

- ✚ Analisar e descrever o processo de alojamento e pagamento de mensalidades de arrendamento de camas para melhor percepção do sistema actual;
- ✚ Desenhar e propor um modelo para o processo de alojamento e pagamento de mensalidades de arrendamento de camas nas residências universitárias;
- ✚ Analisar e escolher o modelo de desenvolvimento de SI;
- ✚ Desenvolver um protótipo que simule a implementação do modelo.

## 4- Estrutura do Trabalho

O presente trabalho está estruturado da seguinte maneira:

Capítulo I - Introdução, faz-se uma breve contextualização, define-se os problemas a serem resolvidos, os objectivos e apresenta-se a estrutura do trabalho.

Capítulo II - Metodologia, apresenta-se a metodologia que foi seguida para se concretizar os diversos objectivos específicos deste trabalho.

Capítulo III - Revisão Bibliográfica, aborda-se de forma superficial alguns modelos utilizados no desenvolvimento de SI com objectivo de posteriormente escolher o modelo julgado ideal para o desenvolvimento de SI proposto, de acordo com o recurso tempo, dinheiro e pessoas.

Capítulo IV - Caso de Estudo: Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM, descreve-se a situação actual, o processo de alojamento e desalojamento, o processo de permuta e troca de quartos e aborda-se de forma resumida algo sobre as residências universitárias.

Capítulo V - Modelo Proposto de Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM, faz-se a descrição do modelo proposto e usa-se diagramas de *Unified Modelling Language (UML)* baseado em casos de usos, diagramas de classes, diagramas de actividades, diagramas de sequência e diagramas de estados para modelação.

Capítulo VI - Conclusões e Recomendações, são apresentadas as conclusões do presente trabalho e recomendações para aspectos que o autor não alcançou ou gostaria de alcançar, mas são fundamentais para aumentar a eficiência do modelo proposto.

Finalmente, apresenta-se as Referências Bibliográficas e os Anexos para melhor sustentar a percepção desta dissertação.

## Capítulo II – Metodologia

Este capítulo aborda de forma detalhada todos os passos que se seguiram para a elaboração do presente trabalho.

O primeiro passo com vista a elaboração da pesquisa foi obter junto à secretária do Departamento de Matemática e Informática (DMI) uma credencial que foi entregue à DSS, como forma de pedido de autorização para que a recolha de dados fosse feita naquela instituição. Para se alcançar os objectivos preconizados, a pesquisa obedeceu a metodologia que abaixo se descreve.

*Para analisar e descrever o processo de alojamento e pagamento de mensalidades de arrendamento de camas para melhor percepção do sistema actual:*

Utilizou-se o caso de estudo como estratégia de pesquisa, pois, a mesma permitiu delimitar o tema a ser pesquisado. Portanto, como técnica de recolha de dados, o candidato usou a triangulação, que é o nome designado quando o trabalho envolve mais da uma técnica de recolha de dados.

Dentre as várias técnicas de recolha de dados, usou-se entrevista, com vista a esclarecer alguns detalhes que não estão documentados e que sejam relevantes ou significativos para a pesquisa e a observação, pois existem aspectos que só ficaram claros observando-os. As entrevistas foram feitas a alguns funcionários da DSS, dentre eles, administradores das RU e funcionários do DA.

Feita a recolha de dados, efectuou-se a análise dos mesmos que incidiu sobre os seguintes documentos: respostas do guião de entrevista, requisitos sistema e documentos referentes a DSS. Esta análise consistiu em comparar (semelhanças e diferenças) as diversas respostas fornecidas pelos entrevistados.

*Para desenhar e propor um modelo para o processo de alojamento e pagamento de mensalidades de arrendamento de camas nas residências universitárias:*

Foi usada uma abordagem orientada a objectos, pois enquadra-se perfeitamente com o tipo de problema presente, aplica-se em várias áreas tecnológicas, pelo facto de ser do domínio do autor e por emprestar várias vantagens tais como: extensibilidade<sup>1</sup>, redução de custos<sup>2</sup>, facilita a comunicação entre desenvolvedores e clientes<sup>3</sup>, melhora a qualidade, a estrutura da base de dados e funcionalidades dos programas podem ser desenhadas no mesmo paradigma, etc.

Tendo sido escolhida uma abordagem orientada a objectos, para modelação usou-se *Unified Modelling Language* (UML) que segundo NUNES e O'NEILL (2003) é uma linguagem de modelação que utiliza a notação padrão para especificar, construir, visualizar e documentar SI orientados a objectos.

Sendo assim, neste trabalho usou-se o *Dia*<sup>4</sup> para incorporar a UML, visto que existe uma necessidade de se produzir os diagramas UML antes da programação começar a ser implementada.

*Para analisar e escolher o modelo de desenvolvimento de SI:*

Fez-se a comparação de alguns modelos de desenvolvimento de *software* a escolha do autor e conforme as vantagens, desvantagens e aplicabilidade de cada modelo, escolheu-se o modelo incremental, pois este é o modelo julgado ideal de acordo com o recurso tempo, pessoas e dinheiro para o desenvolvimento do SI proposto.

*Para desenvolver um protótipo que simule a implementação do modelo:*

Utilizou-se o *ZK framework*, a linguagem de programação java, o SGBD *MySQL* e algumas ferramentas de apoio que serão abordadas mais adiante.

---

<sup>1</sup> É flexível a mudança de requisitos.

<sup>2</sup> Documentação clara ajuda na manutenção e permite o reuso de muitos *softwares*.

<sup>3</sup> Os modelos elaborados podem ser entendidos por ambas as partes (clientes e desenvolvedores).

<sup>4</sup> Ferramenta CASE que permite representar de forma rápida os diversos diagramas UML.

O ZK é um *framework ajax* orientado a eventos, baseado em componentes, *open source*, que permite que os desenvolvedores java criem aplicações *Web* com interfaces ricas, sem *JavaScript* e com poucos conhecimentos de programação (CHEN e CHENG, 2007). Este *framework* possui enumeradas vantagens tais como:

- ✚ Facilidade no desenvolvimento de aplicações *Web*, pois existem mais de duzentos (200) componentes prontos, tais como: *JfreeChart*, *JasperReports*, *Google Maps*, *FCKeditor*, *Timeline*, *ExtJS*, *Dojo* e mais;
- ✚ É *open source* e a comunidade de desenvolvedores ZK é extremamente activa com mais de vinte (20) traduções, mais de cem (100) artigos/blogs e mais de um milhão e trezentos mil (1.300.000) *downloads* de mais de cento e noventa (190) países;
- ✚ Permite o acesso móvel, isto é, o ZK estende o alcance de sua aplicação para os dispositivos móveis;
- ✚ Possui maior segurança porque este foi concebido desde sua criação para ser seguro e as suas aplicações *Web* estão protegidas contra *JavaScript* ou *SQL injection* maliciosos.

O java foi escolhido como linguagem de programação, porque esta é usada hoje em dia, para criar páginas *Web* com conteúdo dinâmico e iterativo, para desenvolver uma larga escala de aplicações empresariais e para aumentar as funcionalidades dos servidores *Web* (DEITEL, H. e DEITEL, P., 1999). Para além disso, também escolheu-se o java pelo facto do ZK ser especificamente para desenvolvedores java e por ser uma linguagem bastante conhecida pelo autor.

Como SGBD utilizou-se o *MySQL*, uma vez que, este é muito rápido, robusto, leve (ocupa menos espaço no disco em relação a outros SGBD tais como: *Oracle*, *Microsoft SQLServer*, etc.), de baixo custo, fácil de configurar e aprender, portátil e permite o armazenamento, a procura, o ordenamento e recuperação de dados de forma eficiente (WELLING e THOMSON, 2001).

Por fim, como ferramentas de apoio usou-se o *Microsoft Office 2007*, o *iReport-3.0.0*, o *Apache Tomcat 6.0*, *Microsoft Office Visio 2007* e o *eclipse*.

O *Microsoft Office 2007* foi usado para escrever os relatórios do trabalho de pesquisa.

O *iReport* foi utilizado para desenhar os diversos relatórios, pelo facto de ser grátis, *open source*, permitir o acesso aos dados e a partir destes gerar relatórios no formato PDF, RTF, XML, HTML, DOCX, etc.

O *Apache Tomcat* foi escolhido como servidor de aplicação, não só pelo facto de ser *open source*, mas também por ser um servidor de aplicação java para *Web* e principalmente por ser mais leve se comparando com outros como Oracle Weblogic, Glassfish, Oc4j e de fácil configuração.

A arquitectura do sistema proposto e o organograma presente no trabalho foram desenhados com ajuda do *Microsoft Office Visio 2007*.

Como ambiente de desenvolvimento integrado (IDE<sup>5</sup>) de desenvolvimento da aplicação *Web*, seleccionou-se o *eclipse* porque nele pode-se facilmente instalar o *ZK framework* como *plug-in*.

---

<sup>5</sup> IDE - do inglês *Integrated Development Environment* ou **Ambiente Integrado de Desenvolvimento**, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de *software* com o objectivo de agilizar este processo.

## Capítulo III – Revisão Bibliográfica

### 1- Introdução

Este capítulo destina-se a revisão bibliográfica<sup>6</sup> de alguns modelos de processo de *software*<sup>7</sup> a escolha do autor, e tem como objectivo analisar e escolher o melhor modelo (ou modelo julgado ideal) de desenvolvimento de SI, que posteriormente será usado para construção do sistema proposto.

O sucesso ou fracasso durante o desenvolvimento de SI depende de diversos factores que ocorrem durante todo o processo de desenvolvimento do *software*, pois, desenvolver um *software* é geralmente uma tarefa complexa que está sujeita a erros. Por este motivo e pelo facto do desenvolvimento de SI seguir um método que engloba um conjunto de regras, etapas e actividades que devem ser satisfeitas, existe uma necessidade de estabelecer-se processos sistemáticos ou modelos de processo de *software* (NUNES e O'NEILL, 2003).

No entanto, apesar de existirem diversos modelos de processo de *software* que devem ser seguidos durante o desenvolvimento de *software*, a maior parte dos grandes sistemas, utiliza diferentes abordagens para diferentes partes do sistema, visto que não existe um modelo ou abordagem ideal de desenvolvimento de *software* (SOMMERVILLE, 2003).

É relevante salientar que a UML pode ser usada em diversas abordagens de desenvolvimento de *software*, começando das abordagens tradicionais (por exemplo: Cascata, Incremental, etc.) que são utilizadas quando os requisitos do sistema a ser desenvolvido são estáveis e os requisitos futuros são previsíveis, até as abordagens mais recentes denominadas abordagens ágeis (exemplo: Programação Extrema, Scrum, etc.) em que os requisitos mudam rapidamente e permitem que a equipe de desenvolvimento

---

<sup>6</sup> Revisão bibliográfica tem como objectivo obter informações sobre a situação actual do tema ou problema, conhecer publicações já existentes sobre o tema, verificar opiniões similares e diferentes a respeito do tema.

<sup>7</sup> Modelo de processo de software – é uma descrição simplificada de um processo de software, que é apresentada a partir de uma perspectiva específica.



se concentre apenas no *software*, em vez de em seu projecto e documentação, com vista a entregar o produto de *software* aos clientes o mais rapidamente possível.

Conforme os objectivos deste trabalho, escolheu-se como modelos para análise os seguintes: modelo de incremental, processo de modelação unificado e a programação extrema. Estes modelos serão abordados de forma superficial e resumida mais adiante.

## **2- Modelo de Desenvolvimento Incremental**

### **2.1- Introdução**

O modelo de desenvolvimento incremental ou simplesmente modelo incremental é uma abordagem que combina as vantagens do modelo cascata, onde as fases são sequenciais e os requisitos devem estar claros no início, e do modelo evolucionário, em que a ideia base é desenvolver uma implementação inicial que se aperfeiçoa por meio de muitas versões e os requisitos podem ser definidos gradualmente (SOMMERVILLE, 2003).

Este modelo foi proposto com objectivo de reduzir o retrabalho no processo de desenvolvimento de *software* e de proporcionar aos clientes algumas oportunidades de adiar decisões sobre os requisitos detalhados do sistema, até que eles tenham alguma experiência com o sistema a ser desenvolvido. Entretanto, para além dos objectivos citados anteriormente, o modelo incremental também têm como objectivo apresentar um produto operacional a cada incremento<sup>8</sup>.

Durante o desenvolvimento do *software*, pode-se realizar uma análise dos próximos requisitos para os incrementos posteriores, mas não são aceitas mudanças de requisitos no incremento actual, ou seja, a cada novo incremento acrescenta-se novas funcionalidades e reforça-se as funcionalidades dos incrementos anteriores (SOMMERVILLE, 2007).

Estes incrementos podem ser desenvolvidos em uma ou mais iterações. Durante estas iterações, os clientes avaliam e identificam as funcionalidades mais e menos

---

<sup>8</sup> Cada incremento fornece um subconjunto das funcionalidades do sistema.

importantes a serem fornecidas pelo sistema, com vista a entregar as funcionalidades prioritárias em primeiro lugar aos clientes, evitando deste modo, que se encontrem falhas nas partes mais importantes do sistema no final do desenvolvimento.

Salientar que, não existe uma necessidade de se utilizar a mesma abordagem de desenvolvimento a cada incremento, uma vez que este modelo é uma combinação de duas abordagens, o que significa que pode se usar o modelo cascata se as funcionalidades de um incremento específico estiverem bem definidas, e o modelo evolucionário caso contrário.

## 2.2- Características

As principais características do modelo de desenvolvimento incremental segundo SOMMERVILLE (2007) são:

- ✚ O sistema é desenvolvido em uma série incrementos;
- ✚ O processo de especificação, projecto e implementação são concorrentes. Este aspecto é inevitável quando há mudanças nos requisitos;
- ✚ Deve se indicar as funcionalidades prioritárias que serão desenvolvidas em primeiro;
- ✚ Alto nível de visibilidade de riscos, uma vez que eles precisam ser minimizados rapidamente para que uma entrega do *software* seja realizada com sucesso.

## 2.3- Vantagens e desvantagens

De acordo com SOMMERVILLE (2007) o modelo desenvolvimento incremental têm uma série de vantagens e desvantagens:

Vantagens

- ✚ Os clientes não precisam esperar até a entrega do sistema inteiro para se beneficiarem dele;
- ✚ Os clientes podem usar os incrementos iniciais como protótipos e ganhar experiência, obtendo deste modo, informações sobre os requisitos dos incrementos dos estágios posteriores do sistema;
- ✚ Existe um menor risco de fracasso completo do sistema;
- ✚ É menos provável que se encontrem erros nas partes mais importantes do sistema, visto que estes são entregues em primeiro e passam pela maior parte dos testes.

#### Desvantagens

- ✚ Os incrementos devem ser relativamente pequenos, isto é, não podem ser compostos por mais de vinte (20) mil linhas de código;
- ✚ É difícil mapear os requisitos do cliente em incrementos de tamanho adequado;
- ✚ Pode ser difícil identificar os recursos comuns exigidos por todos incrementos, visto que os incrementos não são definidos detalhadamente até que o incremento seja implementado;
- ✚ Alguns incrementos podem precisar de ser retirados de uso e retrabalhados, se porventura os requisitos forem instáveis ou incompletos;
- ✚ A gestão de custos, cronograma e configuração é mais complexo.

### **3- Processo de Modelação Unificado**

#### **3.1- Introdução**

Segundo JACOBSON et. al (1999) referenciado por NUNES e O'NEILL (2003), Processo de Modelação Unificado (PMU) é uma abordagem iterativa e incremental que sugere a utilização efectiva da UML.

Actualmente, este processo e a UML são amplamente usados em projectos orientados a objectos de todas as naturezas (*softwares* de médias e grandes dimensões), uma vez que, grande parte dos produtos, como por exemplo, documentos e diagramas podem ser produzidos usando a UML.

O PMU quando representado graficamente (Figura 1), possui duas perspectivas: uma perspectiva dinâmica, que mostra as fases ao longo do tempo e outra estática, que mostra as actividades realizadas durante o processo. Na perspectiva dinâmica são identificadas 4 fases: início, elaboração, construção e transição; e na perspectiva estática são observadas diversas actividades técnicas, tais como: modelação do negócio, levantamento de requisitos, análise e desenho, programação, teste e instalação.

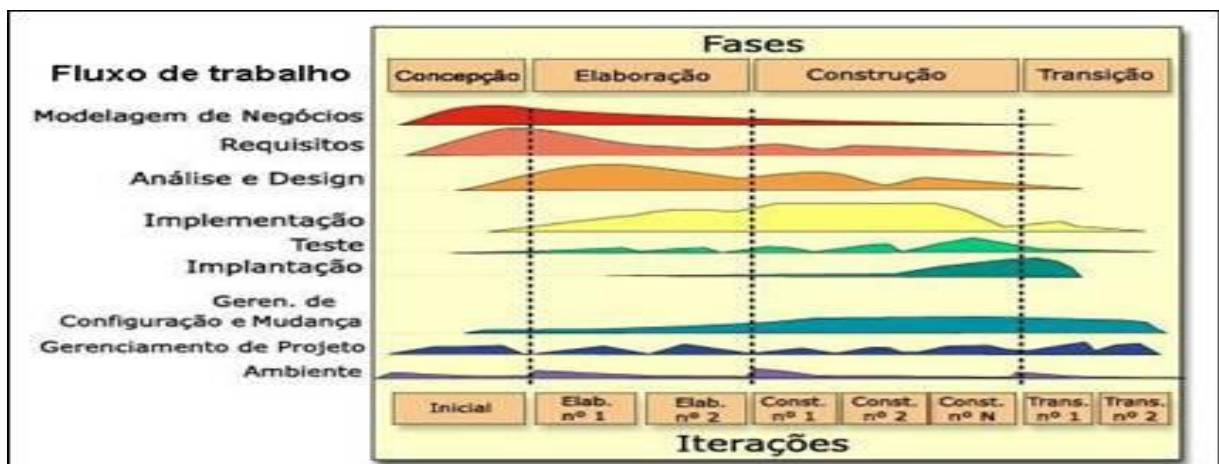


Figura 1- Relação entre as fases e actividades do PMU

### 3.2- Características

Para NUNES e O'NEILL (2003) o PMU possui 3 características básicas que são:

- ✚ Deve ser iterativo e incremental;

Iterativo: deve permitir que o desenvolvimento do SI seja em ciclos sucessivos, pois, segundo SOMMERVILLE (2007) as actividades do processo de *software* são repetidas regularmente à medida que o sistema é retrabalhado, em resposta às solicitações de mudança.

Incremental: deve haver um aumento gradual do conhecimento sobre o domínio da aplicação, das funcionalidades exigidas, e deve permitir a inclusão de novas funcionalidades e melhorias contínuas.

- ✚ Deve ser centrado em casos de uso, de modo a espelhar as funções que sistema deve proporcionar a certo conjunto de utilizadores do sistema (actores);
- ✚ Baseia-se numa arquitectura de modelação, uma vez que, esta abordagem deve permitir caracterizar a estrutura e comportamento do SI, suas funcionalidades, seu nível de desempenho, as interfaces com os utilizadores e outros sistemas, e enquadrar os contributos complementares dos diversos participantes no projecto (programadores, analistas, utilizadores, gestores, etc.);

Para além das características citadas acima, este modelo possui outras características que são:

- ✚ Deve permitir o desenvolvimento de componentes que possam ser programados, testados de forma autónoma e reutilizados em diversos sistemas;
- ✚ Deve permitir a gestão de equipas de dimensão adequada, atribuindo responsabilidades por tarefas que possam ser realizadas em paralelo, como forma de reduzir o ciclo temporal de desenvolvimento;
- ✚ Deve facilitar a elaboração da documentação de utilização e de administração do sistema.

### **3.3- Actividades**

Segundo NUNES e O'NEILL (2003), o PMU possui as seguintes actividades:

- ✚ Modelação do negócio: serve de enquadramento ao SI, visto que, este descreve a estrutura e dinâmica da organização.
- ✚ Levantamento de Requisitos: descreve as características e as propriedades desejadas para o sistema pelos potenciais utilizadores.

- ✚ Análise: descreve o que o sistema deve fazer com rigor, mas sem restrição quanto à natureza técnica da solução que venha ser adoptada.
- ✚ Desenho: descreve a arquitectura do sistema, identificando com um alto nível de detalhe o modo como os requisitos devem ser satisfeitos do ponto de vista técnico.
- ✚ Codificação: é a actividade que se realiza durante o desenvolvimento de SI e testes unitários.
- ✚ Integração e teste: destina-se a integração dos diversos módulos de *hardware* e componentes de *software*, e avalia-se a robustez do sistema recorrendo a métricas de deteção de erros.
- ✚ Instalação: é durante a realização desta actividade que se disponibiliza e se implementa a versão operacional do sistema.
- ✚ Gestão de configuração: destina-se a controlar as mudanças que surgem nos diversos produtos do processo de desenvolvimento de *software*<sup>9</sup> e inclui tarefas de manutenção correctiva e evolutiva.

Para além destas actividades, também deve se realizar algumas actividades de apoio, tais como, gestão de projecto e gestão de mudanças para que se desenvolva um sistema com sucesso.

### 3.4- Fases

Ao longo das fases de desenvolvimento realizam se diversas actividades, existindo actividades dominantes em cada uma delas. Sendo assim, a seguir são apresentadas as 4 fases de desenvolvimento que fazem parte do PMU (NUNES e O'NEILL, 2003):

- ✚ Início: é a primeira fase do PMU e também é denominada fase de Concepção. Nesta fase define-se o âmbito do projecto e suas fronteiras, os critérios de avaliação de sucesso e de risco, a estimativa dos recursos necessários, efectua-se

---

<sup>9</sup> Processo de desenvolvimento de software – é um conjunto de actividades e resultados associados que geram um produto de software.

o levantamento de requisitos e elabora-se um plano de trabalho com as principais etapas, actividades e pontos de controlo. Os resultados esperados nesta fase são: o plano do projecto inicial, diagramas de casos de uso iniciais e por vezes um protótipo simplificado.

- ✚ **Elaboração:** procura analisar em detalhe o domínio do problema, estabelecer uma arquitectura, desenvolver o plano de projecto e eliminar os factores de riscos. Para que estas actividades ocorram com sucesso, deve-se ter uma visão e compreensão de todo o sistema, isto é, o seu âmbito, os requisitos funcionais e não funcionais. Esta fase possui como resultado, para além da descrição da arquitectura do *software* e um (ou vários) protótipo (s) que suportem os principais casos de uso, os mesmos resultados da fase inicial mais com maior nível de detalhe.
- ✚ **Construção:** nesta fase, o produto a ser disponibilizado aos utilizadores é desenvolvido de forma iterativa e incremental até que todos os requisitos sejam atendidos. Nela detalha-se os restantes casos de uso, define-se os critérios de aceitação, refina-se o desenho, efectua-se a codificação e testes de aplicação. Sendo assim, ao concluir esta fase, deve se ter um produto de *software* em funcionamento e a respectiva documentação pronta para ser disponibilizada ao utilizador.
- ✚ **Transição:** é a última fase do PMU, pois é nela que se disponibiliza o produto de *software* aos utilizadores finais e coloca-se o sistema no ambiente real. Ao concluir esta fase, deve-se ter um produto de *software* documentado e funcionando correctamente em seu ambiente operacional. Caso se identifique uma necessidade de obter uma nova versão do sistema, dá-se início a uma nova iteração do ciclo de desenvolvimento.

De acordo com a figura apresentada anteriormente (Figura1), pode se observar que em cada uma das fases do PMU admite-se ocorrência simultânea de várias actividades, ainda que exista uma que é dominante. Assim, por exemplo, na fase de início a principal actividade é a modelação de negócio, seguida do levantamento de requisitos, que se estende até a fase de elaboração. Na fase de elaboração se têm como principal actividade a análise e desenho mas também se pode realizar as actividades de implementação e teste.

### 3.5- Vantagens e desvantagens

Abaixo estão descritas algumas vantagens e desvantagens do PMU:

#### Vantagens

- ✚ Permite que o processo de desenvolvimento seja iterativo e incremental, o que pressupõe a existência de sucessivas iterações de refinamento que se repetem ao longo do tempo até se obter a solução final;
- ✚ Mudanças são administradas com facilidade e ocorre um alto nível de reuso<sup>10</sup>;
- ✚ Em cada uma das fases admite-se ocorrência simultânea de várias actividades, ainda que exista uma actividade seja dominante;
- ✚ Os riscos técnicos são identificados e atribuídos prioridades inicialmente, sendo revistos em cada iteração.

#### Desvantagens

- ✚ Exige experiência por parte da equipa de desenvolvimento;
- ✚ É complexo e trabalhoso para projectos de pequeno porte.

## 4- Programação Extrema

### 4.1- Introdução

A recente evolução da abordagem incremental desenvolvida por KENT BECK em 1999 foi denominada por programação extrema (em inglês *extreme programming* (XP)).

O XP é talvez a mais conhecida e mais usada metodologia ágil de desenvolvimento de *software*, voltada para pequenas e médias equipas, e têm como base a entrega de incrementos de funcionalidades muito pequenas, o envolvimento do cliente no processo

---

<sup>10</sup> Reuso de formulários do projecto, código, planos de teste e conhecimento em geral



de desenvolvimento, a constante melhoria do código e a programação aos pares (SOMMERVILLE, 2007).

Esta abordagem é ideal para projectos em que o cliente não sabe exactamente o que deseja e pode mudar de opinião durante o desenvolvimento do SI, uma vez que com o *feedback* constante é possível adaptar-se a eventuais mudanças nos requisitos.

Os requisitos colhidos a partir desta abordagem são expressos por meio de cenários (chamadas histórias do usuário) e mais tarde são divididas em uma série de tarefas. Antes da codificação os programadores que trabalham em pares escrevem os testes unitários referentes a cada tarefa, pois, parte-se do princípio de que duas cabeças normalmente pensam melhor do que uma.

Finalmente, terminada a codificação é feita a integração do código depois de todos os testes terem sido efectuados com sucesso.

## 4.2- Valores

O XP baseia-se em cinco valores para guiar o desenvolvimento do SI, o trabalho em equipa e relações entre diversos indivíduos que participam no processo de desenvolvimento que são: comunicação, simplicidade, *feedback*<sup>11</sup>, respeito e coragem.

Comunicação: o XP procura encorajar a comunicação entre os envolvidos no processo de desenvolvimento de *software* (programadores, analistas, clientes, gestores, etc.). A constante comunicação entre os desenvolvedores visa sintonizar a equipe XP, e entre desenvolvedores e clientes tem como propósito criar um desenho simples e limpo.

Simplicidade: visa programar apenas o necessário para que um determinado requisito seja satisfeito, isto é, minimizar o código e seleccionar o desenho mais simples possível para atender às necessidades dos clientes, desprezando-se as funcionalidades consideradas desnecessárias, pois, normalmente o que o cliente quer é muito mais simples do que aquilo que os programadores constroem.

---

<sup>11</sup> Contacto incessante com o cliente a respeito do projecto

*Feedback*: durante o desenvolvimento de *software* existe um *feedback* constante sobre o estado do mesmo e as decisões tomadas devem ser visíveis e rápidas, visto que todos envolvidos no projecto devem ter consciência do que está acontecendo. Este valor está relacionado com a comunicação, uma vez que, quanto mais *feedback* houver, mais fácil é a comunicação.

*Coragem*: esta é necessária para garantir que o *feedback* do cliente ocorra com frequência, pois, esta abordagem implicará mudanças constantes no produto de *software*. A coragem é essencial em qualquer projecto porque existem decisões que apesar de difíceis, são necessárias para se concluir o projecto com sucesso.

*Respeito*: é o valor que dá sustentação aos demais, uma vez que saber ouvir, compreender e respeitar os pontos de vista dos outros é essencial para que um projecto seja bem sucedido.

### **4.3- Princípios ou Práticas do XP**

Os valores abordados acima são aplicados em cada um dos doze princípios ou práticas do XP que são as seguintes de acordo com PFLEEGER e ATLEE (2006):

1. *Planeamento*: efectua-se o levantamento dos requisitos a partir das histórias dos usuários, define-se o escopo do projecto, o valor de cada requisito e as estimativas de tempo e recursos que serão necessários para elaboração do projecto. É importante citar que em todo este processo há participação do cliente.
2. *Pequenos incrementos*: o sistema é subdividido em pequenas partes, visto que este deve ser concebido de modo que determinadas funcionalidades sejam entregues o mais depressa possível, expandindo-se gradualmente durante o desenvolvimento.
3. *Metáfora*: visa ajudar a equipe de desenvolvimento a ter um entendimento comum sobre o vocabulário utilizado no domínio do projecto, facilitando deste modo os desenvolvedores a nomearem de forma comum, por exemplo, funções e variáveis.
4. *Projecto simples*: visa implementar somente a funcionalidade que foi solicitada, isto é, o *software* deve ser o mais simples possível e deve satisfazer os requisitos actuais.

5. Desenvolver testando primeiro: deve se escrever os testes unitários para cada nova parte da funcionalidade do sistema antes da sua implementação.
6. Refactoração: deve ser feita sempre que possível visando melhorar o código mas sem perder as funcionalidades actuais. Isto simplifica o código actual e torna-o fácil de manter.
7. Programação em pares: os programadores trabalham em pares e usam um único teclado para desenvolver o sistema. Cada um deles verifica o trabalho do outro e fornece apoio para se efectuar um trabalho com sucesso, apesar de estes possuírem responsabilidades específicas que podem e devem ser alteradas sempre que possível.
8. Propriedade colectiva: incentiva os membros da equipa de desenvolvimento a contribuir com novas ideias em todo segmento do projecto. Qualquer um pode alterar qualquer parte do sistema para adicionar funcionalidades, corrigir erros, melhorar o projecto (refactorar<sup>12</sup>). Geralmente, ao seguir este princípio, pode se encontrar dificuldades de administrar os erros introduzidos quando duas pessoas tentam editar o mesmo módulo.
9. Integração contínua: é extremamente necessária, pelo facto do *software* ser desenvolvido rapidamente e em partes. Estas diferentes partes devem ser integradas e testadas, porque os desenvolvedores precisam trabalhar na versão mais recente e as funcionalidades devem ser entregues ao cliente o mais depressa possível.
10. Ritmo sustentável: o XP assume que não se deve fazer horas extras constantemente e estabelece uma meta de 40 horas por semana de trabalho. Quando há necessidade de se fazer horas extras para se poder cumprir com os prazos de entrega, é sinal de que os prazos ou recursos são insuficientes, consequentemente haverá uma redução da qualidade do código e produtividade.
11. Código padrão: o código deve ser formatado de acordo com normas de codificação e todos devem seguir tais normas. Deste modo, o resultado será um bloco de código que parecerá ter sido escrito por uma única pessoa, e este será consistente, legível e de fácil edição.

---

<sup>12</sup> Refactoração – é um processo que permite melhoria constante do código, pois, normalmente os programadores fazem reestruturação do sistema sem mudar o seu comportamento para remover a duplicação, melhorar a comunicação, simplificar ou adicionar flexibilidade.

12. Cliente on-site: o cliente deve estar presente durante todo o desenvolvimento, trabalhando com os desenvolvedores para determinar os requisitos e fornecer um *feedback* dos testes por ele efectuado.

#### 4.4- Actividades

Segundo PRESSMAN (2006) os princípios ou práticas citadas anteriormente ocorrem no decurso de quatro actividades principais que serão apresentadas a seguir:

Planeamento: esta actividade começa com a criação de cenários que descrevem as características e funcionalidades solicitadas pelos utilizadores para o *software* a ser desenvolvido. Estes cenários são desenvolvidos pelos utilizadores e devem ser atribuídos um determinado peso de acordo com o valor do negócio. Em seguida, a equipa de desenvolvimento discute os cenários e pesos com os clientes. Se um determinado cenário necessita de mais de três semanas de desenvolvimento, este deve ser reescrito de forma particionada e atribuídos novos pesos e custos. Depois de definidas as datas e outros aspectos do projecto, deve-se escolher o modo pelo qual será desenvolvido cada cenário. Existem três modos de desenvolver o SI com base em cenários que são: implementar todos os cenários imediatamente, implementar inicialmente os cenários de maior risco e implementar os cenários com valor mais alto em primeiro. Sendo assim, após a primeira versão do projecto ter sido entregue, calcula-se a velocidade de desenvolvimento<sup>13</sup> para ajudar a estimar as datas de entrega e cronograma para versões subsequentes, e ainda verificar se houve algum prejuízo excessivo, visto que este aspecto serve como base para decidir se, posteriormente, o conteúdo de cada versão ou as datas de entregas serão modificados.

Projecto: fornece directrizes de implementação para uma história (cenário) conforme ela foi escrita, não encoraja projectos de funcionalidades extras (segue o princípio *keep It Simple* – mantenha simplicidade) e recomenda a criação de um protótipo operacional para partes do sistema que são desenvolvidas a partir de uma história considerada difícil. Este protótipo é implementado e avaliado com objectivo de diminuir os riscos quando a implementação verdadeira começar.

---

<sup>13</sup> Velocidade de desenvolvimento – é a quantidade de histórias do cliente implementadas durante a primeira versão.

Codificação: o XP recomenda que se elabore uma série de testes unitários para cada uma das histórias que devem ser incluídas na versão actual antes da codificação. Depois de completado o código, este é submetido aos testes unitários fornecendo assim um *feedback* instantâneo aos desenvolvedores. Durante a codificação tem-se como aspectos-chaves: a programação em pares, o desenvolvimento baseado em testes antes da codificação, a integração contínua e a propriedade colectiva.

Teste: inicia com a criação de testes unitários antes da codificação, visto que, este aspecto constitui um elemento-chave da abordagem XP. Estes testes devem ser implementados e automatizados de modo que possam ser executados de forma fácil e repetidamente. À medida que os testes unitários individuais são organizados, o teste de validação e integração podem ocorrer diariamente. Posteriormente, os testes de aceitação são efectuados pelo cliente e focalizam-se nas características e funcionalidades do sistema como um todo.

#### 4.5- Vantagens e desvantagens

Este modelo de processo de *software* possui suas vantagens e desvantagens, e as mesmas serão apresentadas a seguir:

##### Vantagens

- ✚ Permite e encoraja o *feedback* do usuário, elicitando os requisitos reais do sistema;
- ✚ Inconsistências entre requisitos, projecto e implementação podem ser detectados rapidamente;
- ✚ Há uma divisão de carga de trabalho por todo o ciclo de vida;
- ✚ Permite compartilhar as lições aprendidas melhorando continuamente o processo;
- ✚ Evidências concretas do andamento do projecto podem ser oferecidas durante todo o ciclo de vida.

## Desvantagens

- ✚ Não existe uma avaliação de riscos;
- ✚ A análise de requisitos é informal e com isso pode não ser bem vista pelos clientes, pois, estes poderão se sentir inseguros quanto ao funcionamento do sistema;
- ✚ Refatoração do código pode ser vista como irresponsabilidade e incompetência, visto que, não existe uma preocupação formal na utilização do código;
- ✚ Possui falta de documentação, uma vez que, o XP não dá muita ênfase a burocracias (documentos, formulários, processos, controles rígidos, etc.), concentrando-se na elaboração de documentos e diagramas que facilitem no entendimento e identificação do problema.

## **Capítulo IV – Caso de Estudo: Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM**

### **1- Descrição da Situação Actual**

A UEM detém direitos de propriedade e de posse sobre as residências estudantis, através da DSS. A DSS é um órgão da UEM que foi criado com objectivo de executar a política de acção social, através da prestação de apoios, benefícios e serviços nela compreendidos, de modo a proporcionar aos estudantes melhores condições de estudo.

Actualmente, a DRA envia no início de cada ano lectivo, uma lista de estudantes bolseiros para à DSS. Os estudantes bolseiros (com BC) recém chegados<sup>14</sup> devem dirigir-se ao DA, apresentar o recibo de matrícula e portar duas fotografias que serão anexas as fichas de internamento. De seguida, o funcionário do DA deve confirmar o nome do estudante na lista enviada pela DRA. Depois de confirmado, o estudante deve preencher duas fichas de internamento (veja o Anexo 2), uma vez que, uma das fichas é arquivada no DA e a outra é enviada para a residência onde o estudante residirá.

Neste mesmo período, também se recebem cartas de pedidos de arrendamento de camas, por parte de estudantes rendeiros, que podem ser estudantes que perderam a BC<sup>15</sup> ou estudantes não bolseiros. Se a carta de pedido de arrendamento que é dirigida ao director da DSS tiver parecer positivo, o estudante se concordar com as cláusulas deve assinar o contrato de arrendamento de cama (Anexo 3) que é composto por três documentos: termo de contrato, termo de compromisso e vigência do contrato.

Terminada esta etapa, o estudante recebe uma guia de autorização de ocupação de cama (Anexo 4), um extracto de regulamento de residências e um cartão de refeitório que pode ser de bolseiro ou de não bolseiro.

---

<sup>14</sup> Estudante bolseiro que se apresenta pela primeira vez

<sup>15</sup> Por não possuir um bom aproveitamento pedagógico em conformidade com o regulamento pedagógico e de bolsas

A guia de autorização é entregue ao administrador da residência para a qual o estudante foi indicado e segundo a sua categoria poderá usufruir de vários benefícios:

Os estudantes têm direito a uma cama<sup>16</sup>, água, luz, serviços de limpeza, lavanderia, refeitório, sala de estudo e televisão, sala de máquinas com acesso à *Internet* e assistência médica e social<sup>17</sup>.

Os estudantes com BC, para além dos benefícios que todo o estudante que é autorizado a viver na residência universitária têm, usufrui ainda de um jogo de roupa de cama composto por: uma manta, dois lençóis, uma toalha de banho, uma toalha de rosto, duas fronhas e uma almofada.

Para estudantes bolseiros que por algum motivo tenham deixado de beneficiar da BC passando a usufruir apenas a BR e para os não bolseiros (estudantes rendeiros), o processo descrito no terceiro parágrafo deste capítulo é repetido a cada início do ano lectivo. Estes estudantes (rendeiros) devem efectuar o pagamento através do depósito de 750 MT no Banco Standard Bank, na conta nº 106.004462.100.9, até o dia 10 de cada mês.

Feito isso, lhe é retornado pelo banco, um comprovativo de depósito que deve ser entregue ao administrador da residência, onde o estudante vive ou ao DA. Se entregue ao administrador, este no fim do mês deve encaminhar todos os recibos comprovativos de depósito dos estudantes ao DA, visto que no DA regista-se os pagamentos no mapa de pagamentos que consta no processo do estudante (esta actividade inclui também o registo do número do talão de depósito), de modo a permitir um melhor controlo dos pagamentos das mensalidades, e para posteriormente poder aplicar sanções aos estudantes devedores.

Por sua vez, o DA encarrega-se de entregar os recibos comprovativos de depósito à DSS, especificamente à DAF, como forma de controlar as receitas<sup>18</sup> recebidas para posteriormente financiar nas despesas necessárias que são enfrentadas pela DSS. A DAF retorna os recibos comprovativos de pagamento de mensalidades de arrendamento de cama aos administradores. Este processo pode durar meses, uma vez que, depende do volume de trabalho do tesoureiro que passa os recibos.

---

<sup>16</sup> As residências universitárias tem quartos de 2, 3, 4 e 8 camas;

<sup>17</sup> Em caso de doenças crónicas ou morte, a DSS custeia as despesas de transporte para a província de origem;

<sup>18</sup> Montante total em dinheiro, referente a prestação de serviços.



Finalmente, os administradores entregam os recibos comprovativos de pagamento de mensalidades aos estudantes que efectuaram o pagamento. Este recibo comprovativo serve de confirmação, em caso de necessidade, de o estudante ter efectuado o pagamento de arrendamento de cama referente a um determinado mês.

## 2- Residências Universitárias

Pelo facto de o presente trabalho referir-se sobre residências universitárias, há que se saber o que elas são, como estão distribuídas e estruturadas. Sendo assim, vamos neste ponto formar um conceito acerca de residências universitárias.

As residências universitárias (RU) são moradias como qualquer outra, mas que se destinam exclusivamente ao alojamento de estudantes que frequentam a universidade. A DSS possui sete RU e têm a capacidade de alojar mil duzentos e vinte e dois (1222) estudantes ao todo. Das sete RU, duas alojam estudantes do sexo feminino, duas são mistas e as restantes alojam estudantes do sexo masculino (observe a Tabela 1).

A gestão das RU está encarregue a seus administradores. O administrador da RU é a autoridade máxima da residência, e dentre as suas atribuições constam:

- ✚ Gerir o património do estado (residências e seus bens);
- ✚ Alojjar e acompanhar a estadia do estudante, prestando-lhe todo o apoio necessário, isto é, conceder-lhe uma cama e respectiva roupa, verificar se tem luz no quarto assim como nos espaços comuns e se estes estão em condições de habitabilidade, se o estudante reside e cumpre com as normas do regulamento de alojamento, se os estudantes promovem a convivência harmoniosa respeitando-se uns aos outros e se promovem a melhor maneira de se conservar os bens da residência;
- ✚ Verificar se a limpeza dos espaços comuns a cargo dos funcionários está a ser devidamente realizada;
- ✚ Controlar a assiduidade dos funcionários;

- ✚ Aplicar sanções sob ordem do DA, como por exemplo desalojar os estudantes que: tenham perdido a bolsa, pernoitem mais do que trinta dias fora da residência e os rendeiros que não pagam as respectivas mensalidades;
- ✚ Sensibilizar e aconselhar os estudantes, fazendo-lhes saber das consequências que podem advir de certos comportamentos não compatíveis com o viver em comum.

O número de entrada, a capacidade, a designação e o endereço referente a cada residência universitária pode ser vista com mais detalhes na Tabela 1.

**Tabela 1 – Lista de Endereços das Residências Universitárias**

<b>Ord.</b>		<b>Avenida</b>	<b>Nº da Entrada</b>	<b>Capacidade</b>	<b>Designação</b>
1	Residência 1	Amílcar Cabral	1254	217	Masculina
2	Residência 4	Amílcar Cabral	928	76	Feminina
3	Residência 5	Mão Tse Tung	889	157	Mista
4	Residência 6	Campus Universitário (Tangará)		204	Mista
5	Residência 7			204	Feminina
6	Residência 8	Karll Marx	939	176	Masculina
7	Residência 9	Campus Universitário (Tangará)		88	Masculina

### 3- Processos de Negócio

NUNES e O'NEILL (2003, p.57) define processos de negócio como sendo um conjunto integrado de actividades de uma organização, que procuram satisfazer um determinado objectivo e no qual participam um ou mais actores.

A DSS possui vários processos de negócios, mas aqui são abordados somente os processos de negócio referentes ao DA, que são fundamentais para a realização da pesquisa. A seguir se apresenta os seguintes processos de negócios:

#### 3.1- Processo de Alojamento e Desalojamento

No processo de alojamento dar-se-á prioridade de acesso às RU, aos estudantes com BC e bolsa de alojamento (BA). Portanto, os estudantes rendeiros só poderão ter acesso às residências, se estas não estiverem lotadas ou existirem quartos com camas vagas, isto é, os estudantes rendeiros só podem arrendar as camas que sobrarem após o alojamento de todos os estudantes bolseiros, pois, a DSS tem autorização para arrendar 20% das camas, mas este factor não constitui prioridade pelo facto de ser uma excepção.

Sendo assim, as condições para ter direito às RU são as seguintes:

- ✚ Estar matriculado num dos cursos ministrados na UEM;
- ✚ Ser beneficiário da bolsa de estudos;
- ✚ Aceitar e fazer cumprir o regulamento de alojamento e todas outras normas das residências estudantis;
- ✚ Possuir um bom comportamento pedagógico em conformidade com o aproveitamento pedagógico e de bolsas.

Contudo, os estudantes que vivem nas RU podem perder a qualidade de residente perante as seguintes condições:

- ✚ Perda da BC;

- ✚ Perda da BA;
- ✚ Anulação da matrícula;
- ✚ Conclusão do curso;
- ✚ Expulsão da residência;
- ✚ Abandono da residência sem justificação prévia num período superior a trinta dias;
- ✚ Iniciativa própria;
- ✚ Morte do estudante residente;
- ✚ Não pagamento da renda dentro dos prazos estabelecidos.

De Salientar que, das condições acima, existem aquelas em que o estudante perde definitivamente a qualidade de residente, como por exemplo, o não pagamento das mensalidades dentro dos prazos estabelecidos e a expulsão da residência.

### **3.2- Processo de permuta e troca de quartos**

As permutas de quartos são feitas internamente nas residências. O estudante pode permutar de quarto, desde que imediatamente informe ao administrador da RU onde ele vive e isso não seja provocado por problemas de mau comportamento do estudante. A comissão da residência pode recomendar à DSS a impedir uma determinada permuta que pode constituir transferência de um problema de um quarto para outro.

A troca de quartos ou mudança de uma residência para outra requer autorização do DA. Para se efectuar esta troca, o estudante deve apresentar os motivos que o fazem assim desejar ao DA, porque há uma série de procedimentos que devem ocorrer a nível do departamento, como transferir o processo do estudante dos processos de uma residência para outra. Também se realiza um procedimento semelhante em relação a lista de residentes de ambas as residências que participam no processo de mudança.

## 4- Organograma

A figura abaixo apresenta o organograma da DSS, que é o objecto de estudo do presente trabalho, com objectivo de ilustrar graficamente a hierarquia da mesma, ou seja, ilustrar os departamentos, repartições, sectores, cargos e relações entre eles, como forma de visualizar a organização como um todo.

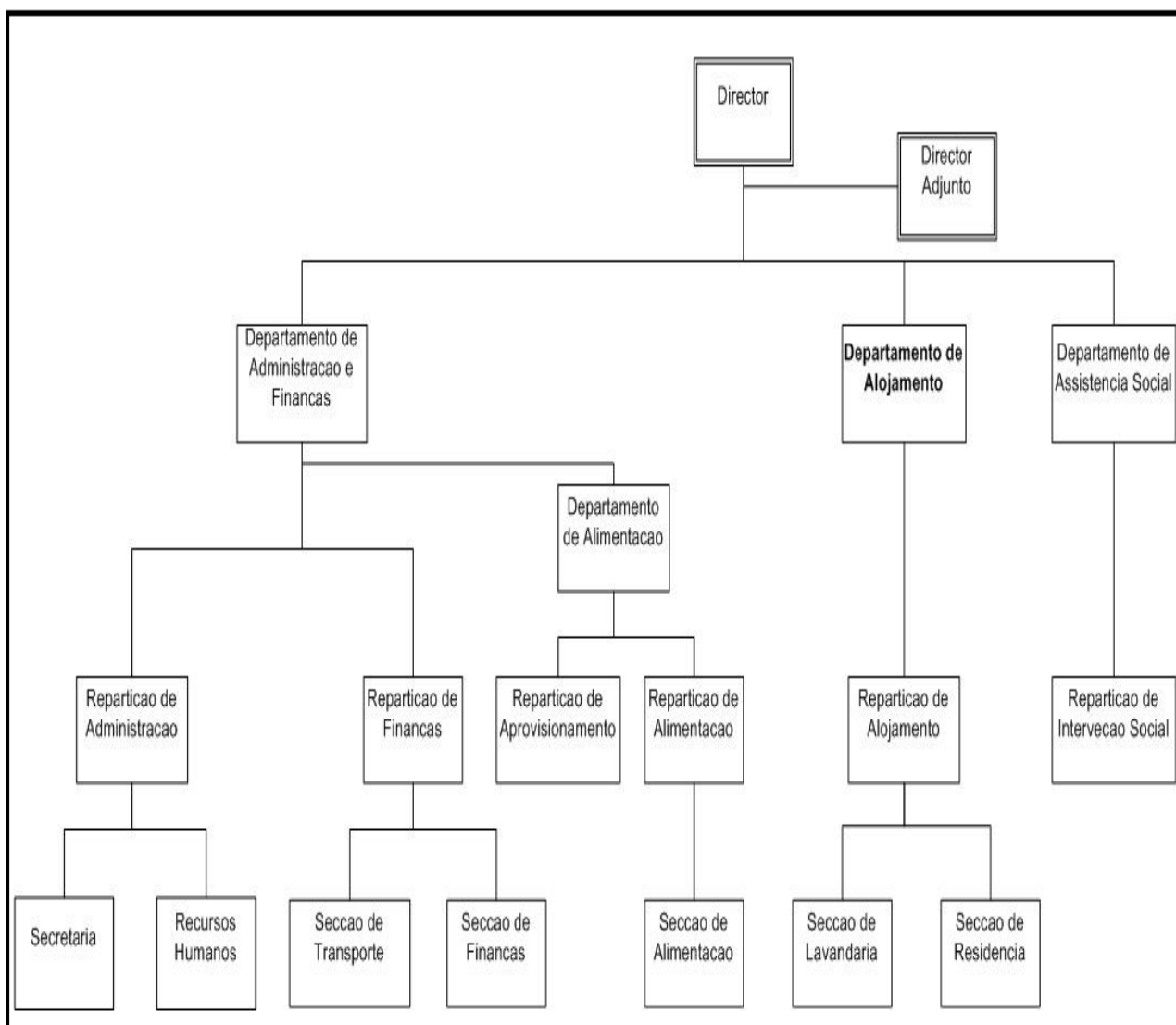


Figura 2 – Organograma da DSS

## **Capítulo V – Modelo Proposto de Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM.**

### **1- Descrição do Modelo Proposto**

As organizações sentem uma necessidade de gerir o recurso informação de forma a garantir uma utilização efectiva e eficiente da mesma, pois, uma correcta gestão da informação permitirá que à organização tenha acesso a informação atempadamente e destrua na altura ideal a informação sem valor (RODRIGUES, 2002).

De forma a garantir que, a DSS tenha acesso a informação em tempo útil, propôs-se desenvolver um modelo que permite efectuar a gestão dos alojamentos nas residências universitárias da UEM com recurso a tecnologias *Web*, uma vez que, a informação fica centralizada numa base de dados (BD) e pode ser acedida e utilizada por todos os níveis de gestão que desejam (e possam) utilizar a mesma em tempo útil (Figura 4).

Deste modo, qualquer actualização na BD far-se-á sentir-se em todos departamentos, secções e repartições que precisam desta para tomar decisões. Sendo assim, o problema de obtenção de informação actualizada e elaboração de estatísticas estará solucionado, visto que todos os dados referentes aos estudantes, como por exemplo, número do estudante, número do bilhete de identidade (BI), nome do estudante, data e local de nascimento, sexo, ano de ingresso e outros estarão centralizados e poderão ser disponibilizados diversos relatórios sobre a situação das RU.

Mas para que tal aconteça, durante o registo de um estudante, deve-se registar para além dos dados do estudante, dados referentes ao seu encarregado de educação (número do BI, nome, profissão, local de trabalho, endereço, telefone, etc.), faculdade (nome, localização), curso (nome, faculdade que o curso pertence) e a residência onde o mesmo tiver sido alocado, se por acaso estes elementos ainda não estiverem registados no sistema.

O fluxo de informação do modelo proposto é similar ao fluxo de informação que se segue actualmente, diferenciando-se nos pontos que serão descritos abaixo e pelo facto do novo modelo ser suportado por TI, cuja importância para as organizações é hoje em dia inquestionável (RODRIGUES, 2002).

Neste modelo, a cada início do ano lectivo, o DA envia um documento sobre a disponibilidade de alojamento por si suportada em cada residência à DRA e tanto os administradores das RU como os funcionários do DA podem efectuar registos de pagamentos de mensalidades referentes ao arrendamento de camas, com vista a aumentar a eficiência e rapidez do processo de pagamentos de mensalidades, uma vez que ambos podem aceder a informação sobre os pagamentos em tempo útil.

Depois do dia 10 de cada mês confirmar-se-á os pagamentos efectuados no DAF, comparando os recibos comprovativos de depósito do banco que foram encaminhados pelo DA com a lista de estudantes que pagaram a mensalidade referente a aquele mês que será apresentada sob forma de relatório no sistema.

Os usuários ou utilizadores do sistema podem efectuar o registo ou actualização de novos estudantes, pagamentos de mensalidades mediante uma autenticação baseada em palavra-chave e se possuem privilégios para efectuar estas operações. Estes privilégios variam de acordo com a categoria do funcionário que pode ser administrador da RU, administrador do sistema, funcionário do DA, funcionário do DAF e director da DSS.

O método de autenticação citado acima, baseia-se num par ordenado constituído pelo código do usuário e por uma senha de acesso associada, pois, segundo MAMEDES (2006) as senhas de acesso constituem a primeira linha de defesa em qualquer sistema computacional que é usado em ambiente de múltiplos utilizadores. Salientar que, estas senhas serão criptografadas usando o método de criptografia<sup>19</sup> *Message Digest Algoritm 5<sup>20</sup>* (MD5) com objectivo de garantir confidencialidade<sup>21</sup>. Após 3 tentativas de login inválidas, o usuário será bloqueado.

Esta informação sobre o bloqueio do usuário fica armazenada no sistema, visto que, é de grande valia para o administrador do sistema. A partir desta informação, o

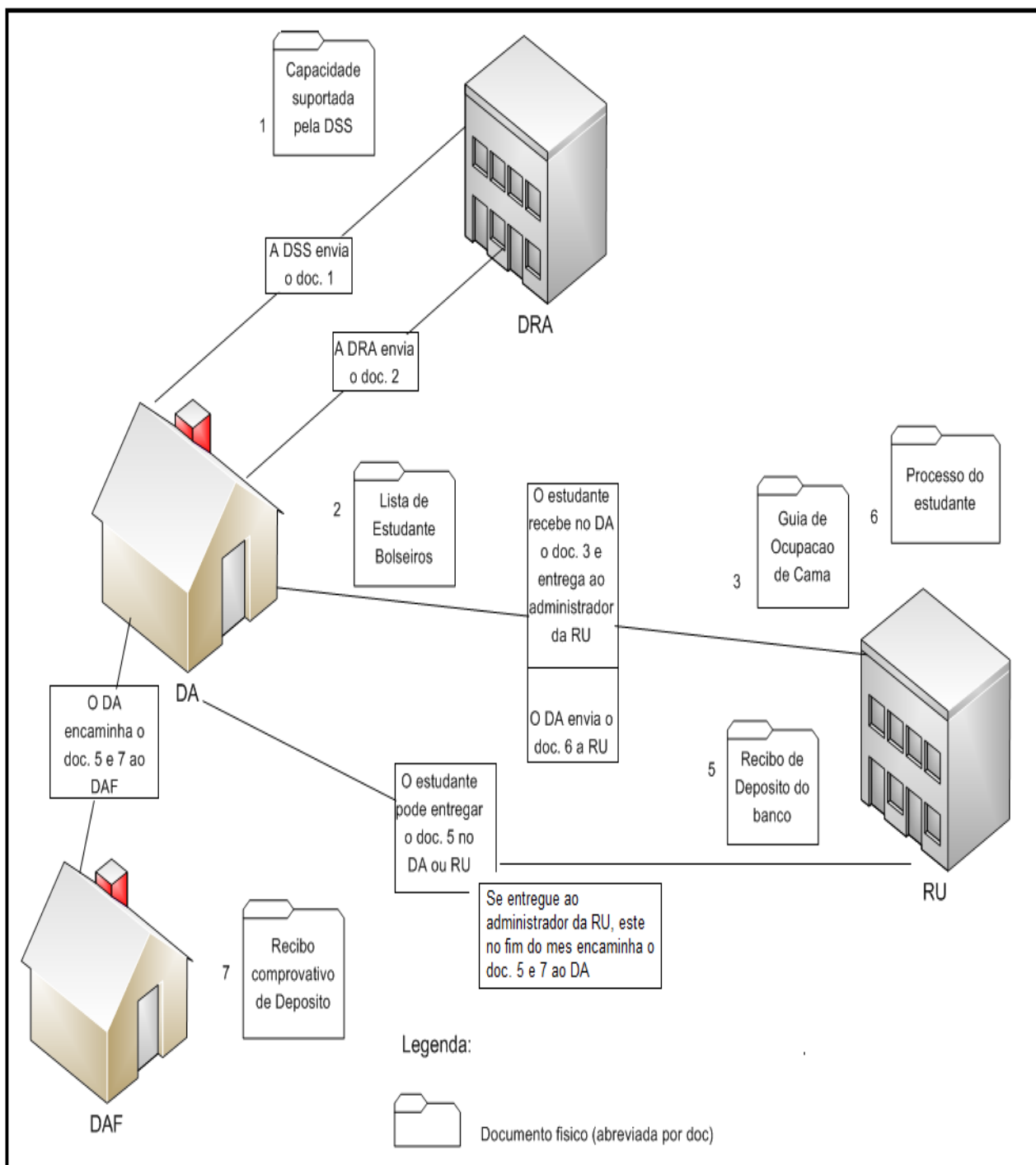
---

<sup>19</sup> Criptografia – é a arte ou ciência de transformar informação recorrendo a chaves secretas, ou seja, é a arte que permite transformar mensagens em claro numa forma não legível, escondendo desta forma o seu conteúdo.

<sup>20</sup> MD5- é um algoritmo hash de 128 bits unidireccional desenvolvido pela RSA Data Security, Inc., usado por softwares de protocolo ponto-a-ponto (P2P) na verificação de integridade de arquivos e logins.

<sup>21</sup> Confidencialidade – está relacionada com a prevenção da utilização não autorizada.

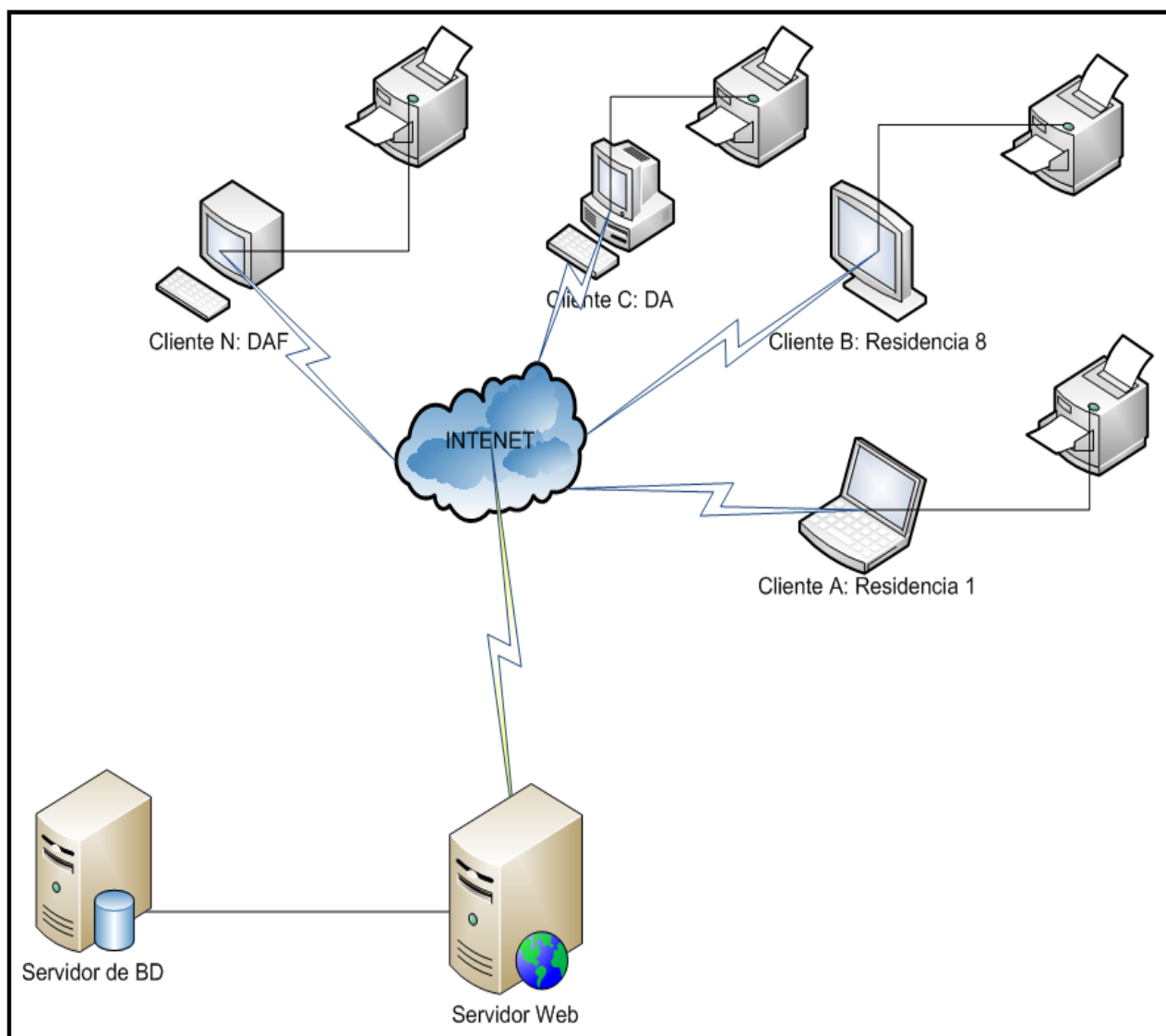
administrador do sistema pode saber se realmente foi o usuário com privilégios para tal, que tentou aceder a aplicação *Web* e deste modo tomar medidas de segurança.



**Figura 3 - Fluxo de informação do modelo proposto**



## 2- Arquitectura do Modelo Proposto



**Figura 4 - Arquitectura do modelo proposto**

## 3- Funcionamento do Modelo Proposto

O primeiro contacto do estudante com o modelo proposto inicia quando o estudante dirige-se ao DA e apresenta os documentos necessários para poder beneficiar-se dos recursos disponibilizados pela DSS.

Após, o estudante apresentar-se ao DA, o modo de proceder diferencia-se um pouco, pelo facto de existirem dois tipos de estudantes que são: bolseiro e rendeiro.

- ✚ O estudante bolseiro (com BC ou BA) recém-chegado preenche as fichas de internamento. De seguida, o funcionário do DA confirma o nome do estudante na lista (em formato físico ou papel) enviada pela DRA. Se o nome do estudante consta na referida lista, o funcionário do DA regista-o na aplicação *Web*.
- ✚ O estudante rendeiro preenche o anexo 3, especificamente o termo de compromisso, e logo em seguida os seus dados são actualizados na aplicação *Web*, se o estudante estiver registado na BD da DSS (estudante que perdeu a BC). Caso o estudante não esteja registado na BD da DSS, os seus dados pessoais, de ingresso a UEM e do encarregado de educação são registados na aplicação *Web*.

Durante o registo do estudante usar-se-á como número único de identificação do estudante, o seu número de estudante, para permitir que este seja encontrado com facilidade em qualquer operação relacionada a UEM, caso haja necessidade de integração com outros sistemas da universidade.

Depois do registo do estudante, aloca-se o estudante num determinado quarto de uma residência específica e emite-se uma guia de autorização de ocupação de cama sob forma de relatório que será entregue ao estudante para posteriormente ser entregue ao administrador da RU onde este tiver sido alojado.

O acto de pagamento de mensalidades referente ao arrendamento de camas na residência ou no DA inicia quando o estudante entrega o talão de depósito do banco, visto que, o talão confirma que este pagou a mensalidade referente ao mês em questão. Em seguida, depois do funcionário da DSS receber o talão e verificar se o estudante existe no sistema, os dados referentes ao pagamento de mensalidades (a data do pagamento, o mês a que o pagamento se destina, o estudante que efectuou o pagamento, o número do talão de depósito e o usuário que efectuou o registo do pagamento) são registados no sistema.

Após o registo dos pagamentos são impressos dois recibos comprovativos de pagamento. O primeiro (original) é entregue ao estudante que pagou a referida mensalidade e o segundo (cópia) é guardado e anexado ao recibo de depósito que foi entregue ao administrador da RU ou ao funcionário do DA para confirmação posterior no DAF.

## 4- Modelação do Modelo Proposto

### 4.1- Descrição dos Actores

Segundo NUNES e O'NEILL (2003), um actor representa uma entidade externa que interage com o sistema, como por exemplo, pessoas e outros sistemas físicos ou lógicos. Sendo assim, a seguir apresentar-se-á a descrição dos actores que poderão interagir com o sistema proposto:

- ✚ **Administrador da RU<sup>22</sup>**: funcionário da DSS que administra uma determinada RU;
- ✚ **Administrador do Sistema**: pessoa que é responsável por administrar a aplicação *Web*;
- ✚ **Director da DSS**: funcionário da DSS que é responsável pela tomada de decisões na DSS;
- ✚ **Funcionário do DA**: trabalhador da DSS que pertence ao DA e possui a responsabilidade de planificar e requisitar as necessidades de alojamento, cobrar os estudantes rendeiros que não efectuaram os pagamentos referentes ao arrendamento de cama, alojar os estudantes que constam na lista enviada pelo DRA e efectuar a confirmação dos estudantes para fins de emissão de cartões de refeitório;
- ✚ **Funcionário da DAF**: trabalhador da DSS que pertence ao DAF e possui a responsabilidade de comprar as necessidades de alojamento requisitadas pelos funcionários do DA e participar no processo de controlo dos pagamentos referentes ao arrendamento de cama efectuados pelos estudantes rendeiros.

---

<sup>22</sup> Autoridade máxima na residência universitária.

## 4.2- Casos de Uso

Tomando como referência, cada um dos actores, nesta secção identificam-se os casos de uso nos quais cada um deles participa:

### 4.2.1- Administrador da RU

- ✚ Efectuar o login;
- ✚ Registar pagamentos;
- ✚ Consultar lista de estudantes;
- ✚ Consultar lista de pagamentos;
- ✚ Permutar o estudante de quarto.

### 4.2.2- Administrador do Sistema

- ✚ Efectuar o login;
- ✚ Registar usuários;
- ✚ Consultar lista de usuários.

### 4.2.3- Director da DSS

- ✚ Efectuar o login;
- ✚ Consultar diversos relatórios sobre as RU.

### 4.2.4- Funcionário do DA

- ✚ Efectuar o login;
- ✚ Registar o estudante;

- + Actualizar estudante;
- + Registrar encarregado de educação;
- + Registrar residência;
- + Registrar faculdade;
- + Registrar curso;
- + Registrar pagamentos;
- + Registrar andar;
- + Registrar quarto;
- + Consultar lista de estudantes;
- + Consultar lista de pagamentos;
- + Consultar lista de residências;
- + Trocar o estudante de residência.

#### **4.2.5- Funcionário da DAF**

- + Efectuar o login;
- + Consultar lista de pagamentos.

### 4.3- Diagramas de Casos de Uso

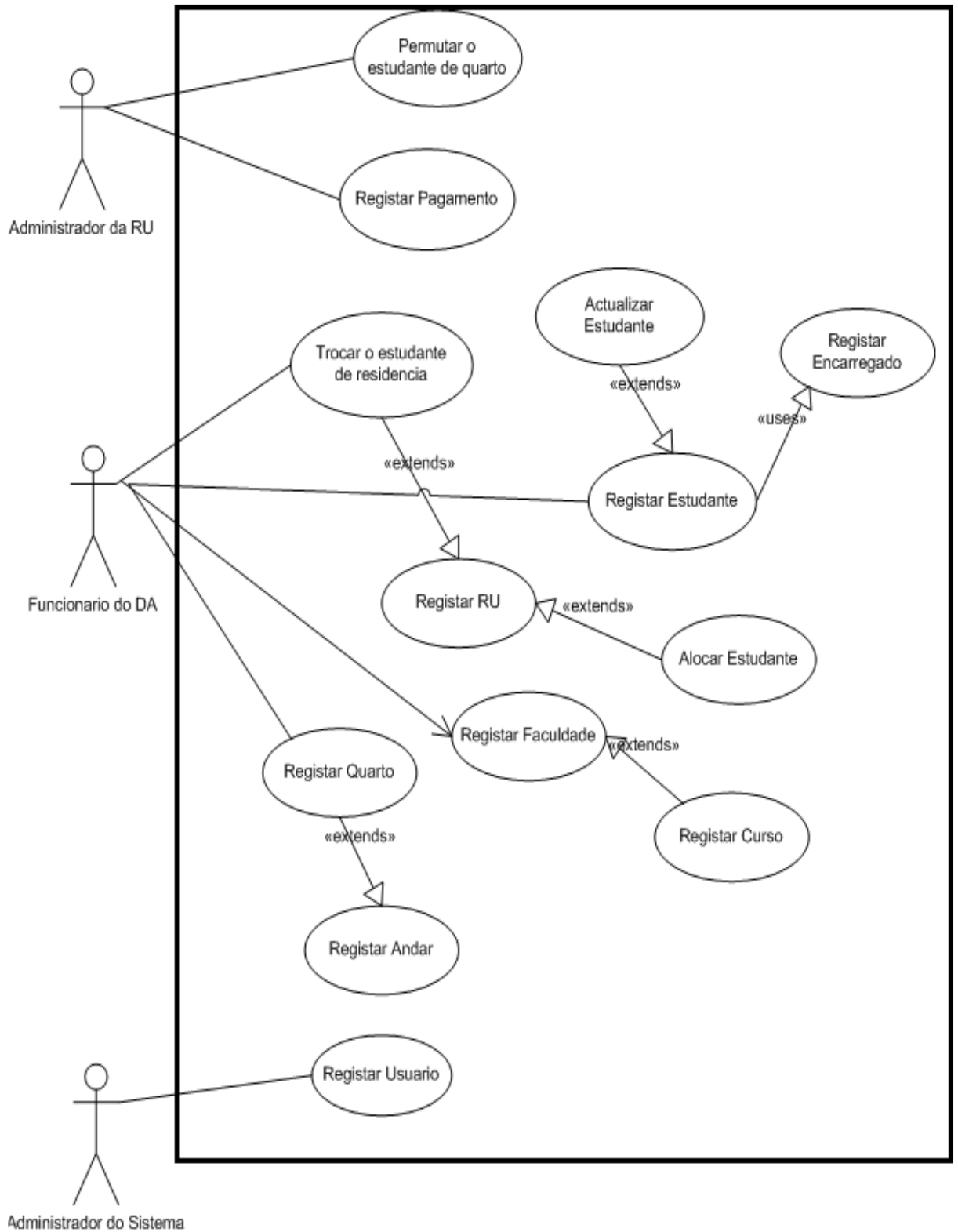


Figura 5 – Diagrama de Casos de Uso para actores emissores (1/2)

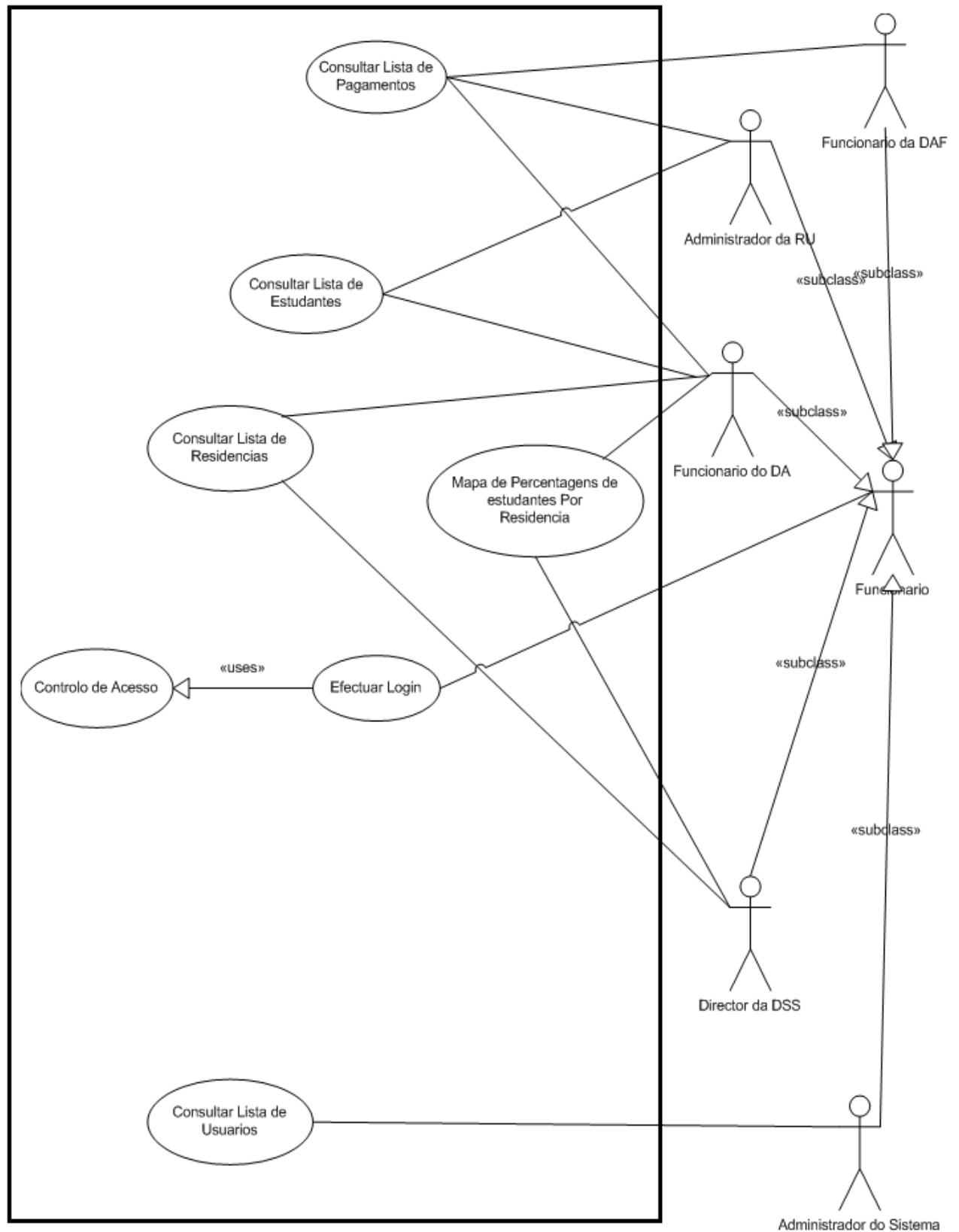


Figura 6 – Diagrama de Casos de Uso para actores receptores (2/2)

#### 4.4- Descrição Estruturada dos Casos de Uso

Nesta secção são descritos alguns casos de uso que serão posteriormente detalhados através de diagramas de sequência na secção 4.6.

<b>Efectuar Login</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema.
<b>Descrição</b>	<ol style="list-style-type: none"><li>1. O caso de uso começa quando o funcionário pressiona o botão Login, após ter digitado o seu <i>username</i> (nome do usuário) e senha de acesso.</li><li>2. O funcionário loga ao sistema pressionado o botão Login.<ol style="list-style-type: none"><li>a) Caso o nome do usuário ou a senha de acesso sejam inválidas, o sistema avisa com uma mensagem (extends Controlo de Acesso).</li></ol></li></ol>
<b>Pós-condição</b>	O sistema disponibiliza o ambiente de trabalho do usuário e mostra-lhe somente o que ele tem direito, omitindo-lhe as restantes funcionalidades.

<b>Controlo de Acesso</b>	
<b>Pré-Condição</b>	O usuário tenta efectuar o login no sistema.
<b>Descrição</b>	<ol style="list-style-type: none"><li>1. O caso de uso começa quando o funcionário pressiona o botão Login após ter digitado o nome do usuário e senha de acesso.</li><li>2. O sistema verifica na sua base de dados a veracidade dos dados introduzidos.</li><li>3. O funcionário dispõe de três tentativas. A 3<sup>a</sup> tentativa falhada o sistema bloqueia a conta do usuário.</li></ol>
<b>Pós-condição</b>	O sistema aprova ou reprovava a autenticação do usuário.

<b>Registar Usuário</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios de administrador do sistema.



<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Administração→Gestão de Utilizadores→Criar Contas de Usuários.</li> <li>2. Em seguida são introduzidos os dados do usuário (BI, nome do funcionário, sexo, nome do usuário, e-mail, senha de acesso, etc.) e escolhe-se a sua categoria.</li> <li>3. O funcionário regista os dados do usuário no sistema pressionando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

<b>Registar Estudante</b>	
<b>Pré-condição</b>	<p>O funcionário é um usuário válido no sistema e possui privilégios para efectuar esta operação.</p> <p>O estudante entrega a ficha de internamento (ou dados que aparecem no contrato de arrendamento) devidamente preenchida ao funcionário do DA.</p>
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→Estudantes.</li> <li>2. O funcionário digita correctamente no sistema os dados do estudante, os dados de ingresso a UEM e os dados do encarregado de educação que aparecem na ficha de internamento.</li> <li>3. O funcionário regista os dados do estudante no sistema pressionando o botão Registrar. <ol style="list-style-type: none"> <li>a) Caso o estudante já esteja registado, o sistema mostra uma mensagem de aviso informando lhe o que esta acontecendo: Extends Actualizar Estudante.</li> </ol> </li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

<b>Registar Residência</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para efectuar esta operação.

<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→Residência.</li> <li>2. Automaticamente o sistema mostra o código da residência a ser registada.</li> <li>3. De seguida, o funcionário introduz a descrição, localização e selecciona o tipo de residência.</li> <li>4. Por fim, o funcionário regista os dados referentes a residência pressionando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

### Registrar Pagamentos

<b>Registrar Pagamentos</b>	
<b>Pré-condição</b>	<p>O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.</p> <p>O estudante entrega o recibo comprovativo de depósito do banco.</p>
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Contabilidade→ Pagamentos.</li> <li>2. O funcionário digita o número do estudante que pretende efectuar o pagamento e clica no botão Pesquisar.</li> <li>3. Feito isto, e se o estudante estiver registado no sistema, automaticamente aparecerá o seu nome e o tipo de bolsa que o mesmo possui.</li> <li>4. De seguida o estudante selecciona a data de pagamento e as linhas contendo o (s) mês (es) a que o pagamento se refere.</li> <li>5. Em cada linha seleccionada deve-se preencher os campos em branco, com o número do talão, valor depositado e banco em que se efectuou o depósito.</li> <li>6. O funcionário confirma o pagamento pressionando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna dois recibos comprovativos de pagamento de mensalidade referente (s) ao (s) mês (es) em que o estudante pagou.

### Registrar Faculdade

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→ Gerais→Faculdade.</li> <li>2. Automaticamente o sistema mostra o código da faculdade a ser registada.</li> <li>3. O funcionário introduz nome e localização da faculdade.</li> <li>4. Por fim, o funcionário regista os dados referentes a faculdade pressionando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

### Registar Curso

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→ Cursos.</li> <li>2. O funcionário introduz o código, nome do curso e selecciona a faculdade que o curso pertence. <ol style="list-style-type: none"> <li>a) Caso a faculdade não esteja registada: Extends Registar Faculdade.</li> </ol> </li> <li>3. De seguida, os dados referentes ao curso são registados no sistema pressionando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

### Registar Andar

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Secundárias→ Andares.</li> <li>2. Automaticamente o sistema mostra o código do andar a ser registado.</li> <li>3. O funcionário selecciona a residência do tipo prédio na qual o andar</li> </ol>

	<p>se localizará e automaticamente aparece a descrição do andar.</p> <p>4. De seguida, introduz-se a lotação do andar, isto é, o número de quartos por andar.</p> <p>5. Finalmente, o funcionário regista os dados referentes ao andar clicando o botão Registrar.</p>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

<b>Registrar Quarto</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Secundárias→ Quartos.</li> <li>2. Automaticamente o sistema mostra o código do quarto a ser registado.</li> <li>3. O funcionário selecciona a residência, andar (se a residência for do tipo prédio) e introduz a descrição e lotação do quarto.</li> <li>4. De seguida, os dados referentes ao quarto são registados no sistema clicando o botão Registrar.</li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

<b>Consultar Lista de Estudantes</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Relatórios→ Listagem→Estudantes→Escolhendo Parâmetros.</li> <li>2. O funcionário selecciona os parâmetros através do qual o relatório do (s) estudante (s) será gerado.</li> <li>3. Finalmente, o funcionário pressiona o botão Gerar Relatório e lhe é retornado a lista de estudante de acordo com os parâmetros</li> </ol>

	<p>seleccionados.</p> <p>a) Se não existe nenhum estudante com os parâmetros seleccionados o relatório é mostrado em branco.</p>
<b>Pós-condição</b>	O sistema retorna a lista de estudantes no formato seleccionado pelo funcionário.

### Consultar Lista de Residências

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Relatórios→ Listagem→Residências→Localização.</li> <li>2. Em seguida lhe é mostrada a lista de residências e as respectivas localizações.</li> </ol>
<b>Pós-condição</b>	O sistema retorna a lista de residências em formato pdf.

### Consultar Lista de Pagamentos

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e possui privilégios para realizar esta operação.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Relatórios→ Listagem→Pagamentos.</li> <li>2. O funcionário escolhe os parâmetros que deseja listar (pagamentos em atrasos, dum determinado mês, etc.).</li> <li>3. Por fim, o funcionário confirma o pedido de listagem pressionando o botão Gerar Relatório.</li> </ol>
<b>Pós-condição</b>	O sistema retorna a lista de pagamentos em formato pdf.

### Permutar o estudante de quarto

<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e pertence a categoria administrador da RU.
---------------------	--

	O estudante solicitou a permuta de quarto por algum motivo.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→Permutar Estudante de Quarto.</li> <li>2. O funcionário introduz o número do estudante e pressiona o botão Pesquisar.</li> <li>3. De seguida, automaticamente o sistema mostra o nome do estudante possui o número digitado e a residência que ele habita.             <ol style="list-style-type: none"> <li>a) Se o estudante não existe, o sistema avisa com uma mensagem.</li> </ol> </li> <li>4. O funcionário selecciona o andar (se a residência for do tipo prédio), o quarto que o estudante deseja e não esteja lotado e a data de alocação.</li> <li>5. Por fim, o funcionário actualiza o quarto do estudante pressionando o botão Permutar.             <ol style="list-style-type: none"> <li>a) Caso, o estudante encontre-se no estado passivo, o sistema informa com uma mensagem.</li> </ol> </li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso.

### Trocar o estudante de residência

<b>Pré-condição</b>	<p>O funcionário é um usuário válido no sistema e pertence a categoria funcionário do DA.</p> <p>O estudante solicitou a troca de residência por algum motivo.</p>
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→Trocar Estudante de Residência.</li> <li>2. O funcionário introduz o número de estudante e pressiona o botão Pesquisar.</li> <li>3. Automaticamente o sistema mostra o nome do estudante, a residência, andar e quarto em que o estudante habita.</li> <li>4. De seguida, o funcionário selecciona no sistema, a residência que o estudante deseja e o aloca em um quarto que não esteja lotado.</li> <li>5. Por fim, o funcionário actualiza a residência do estudante pressionando o botão Registrar.             <ol style="list-style-type: none"> <li>a) Caso, o estudante encontre-se no estado passivo, o sistema informa</li> </ol> </li> </ol>

	com uma mensagem.
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso e emite o guião de autorização de ocupação de cama no formato pdf.

<b>Alojar o estudante</b>	
<b>Pré-condição</b>	O funcionário é um usuário válido no sistema e pertence a categoria funcionário do DA. O estudante deve estar registado no sistema.
<b>Descrição</b>	<ol style="list-style-type: none"> <li>1. O caso de uso começa quando o funcionário selecciona a opção Parametrizações→Gerais→ Alocar Estudante.</li> <li>2. O funcionário introduz o número do estudante e pressiona o botão Pesquisar.</li> <li>3. Automaticamente o sistema mostra o nome do estudante que possui o número introduzido.             <ol style="list-style-type: none"> <li>a) Caso o estudante não esteja registado, o sistema avisa com uma mensagem.</li> </ol> </li> <li>4. O funcionário selecciona no sistema a residência e o quarto que o estudante será alocado.</li> <li>5. Por fim, o funcionário regista os dados referentes ao alojamento do estudante pressionando o botão Registrar.             <ol style="list-style-type: none"> <li>a) Caso, o estudante encontre-se no estado passivo, o sistema informa com uma mensagem.</li> </ol> </li> </ol>
<b>Pós-condição</b>	O sistema retorna uma mensagem informativa referente a operação realizada com sucesso e emite o guião de autorização de ocupação de cama no formato pdf.

## 4.5- Diagrama de Classes

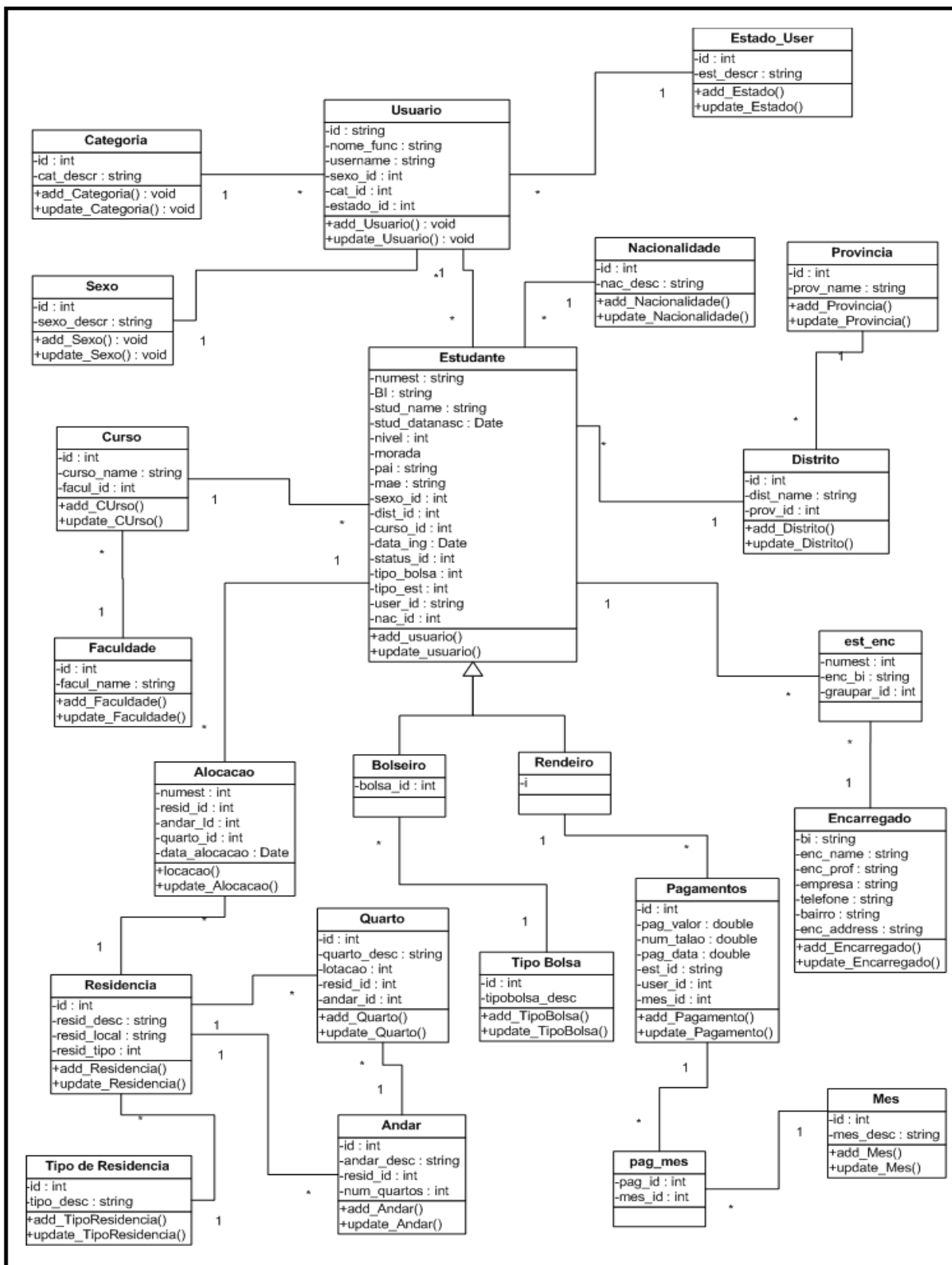
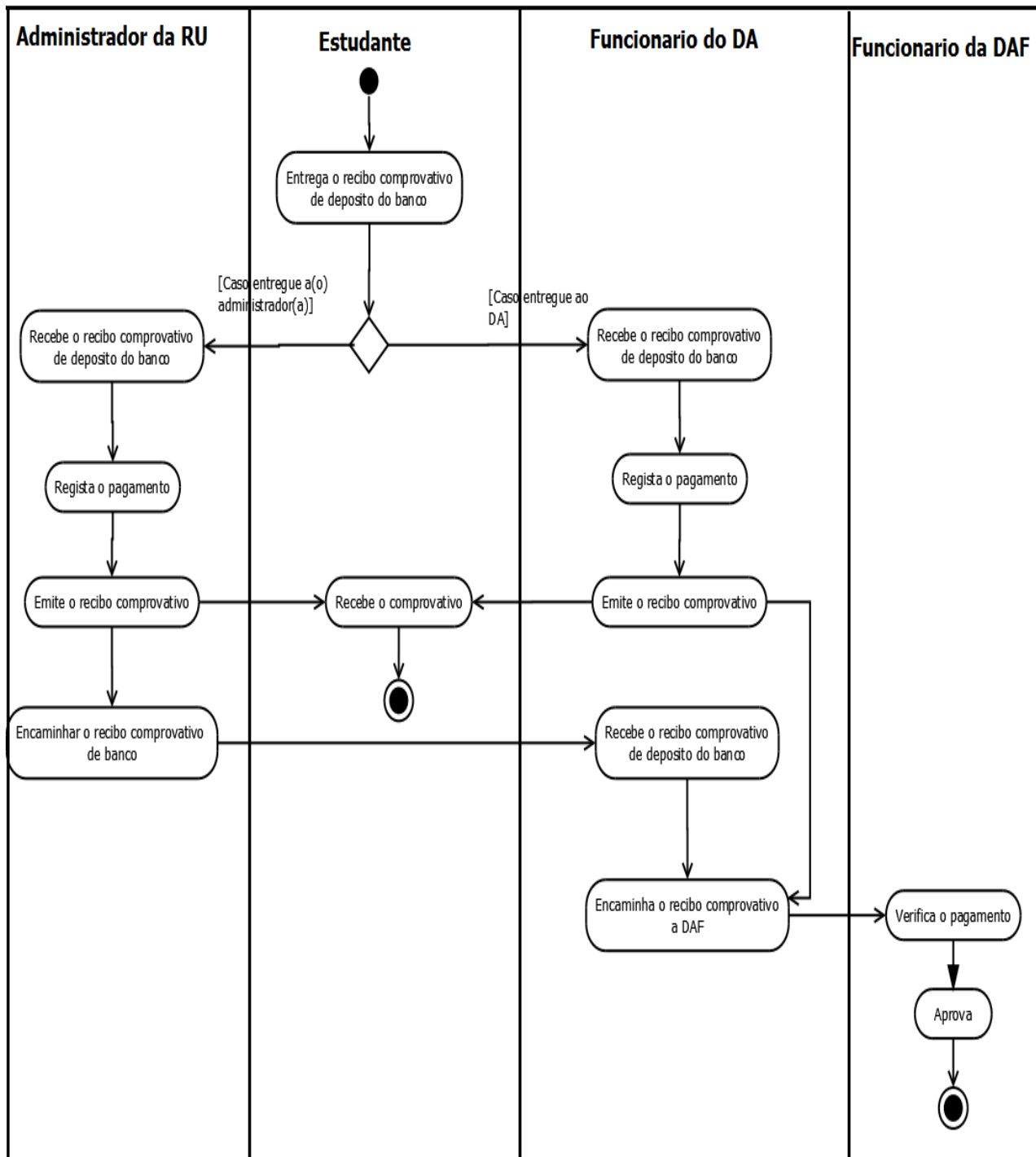


Figura 7 – Diagrama de Classes



## 4.6- Diagrama de Actividades

Nesta secção são apresentadas alguns diagramas de actividades, pois, estes são eficazes para descrever o fluxo de trabalho de uma instituição.



**Figura 8 – Diagrama de Actividades: Registrar Pagamentos**

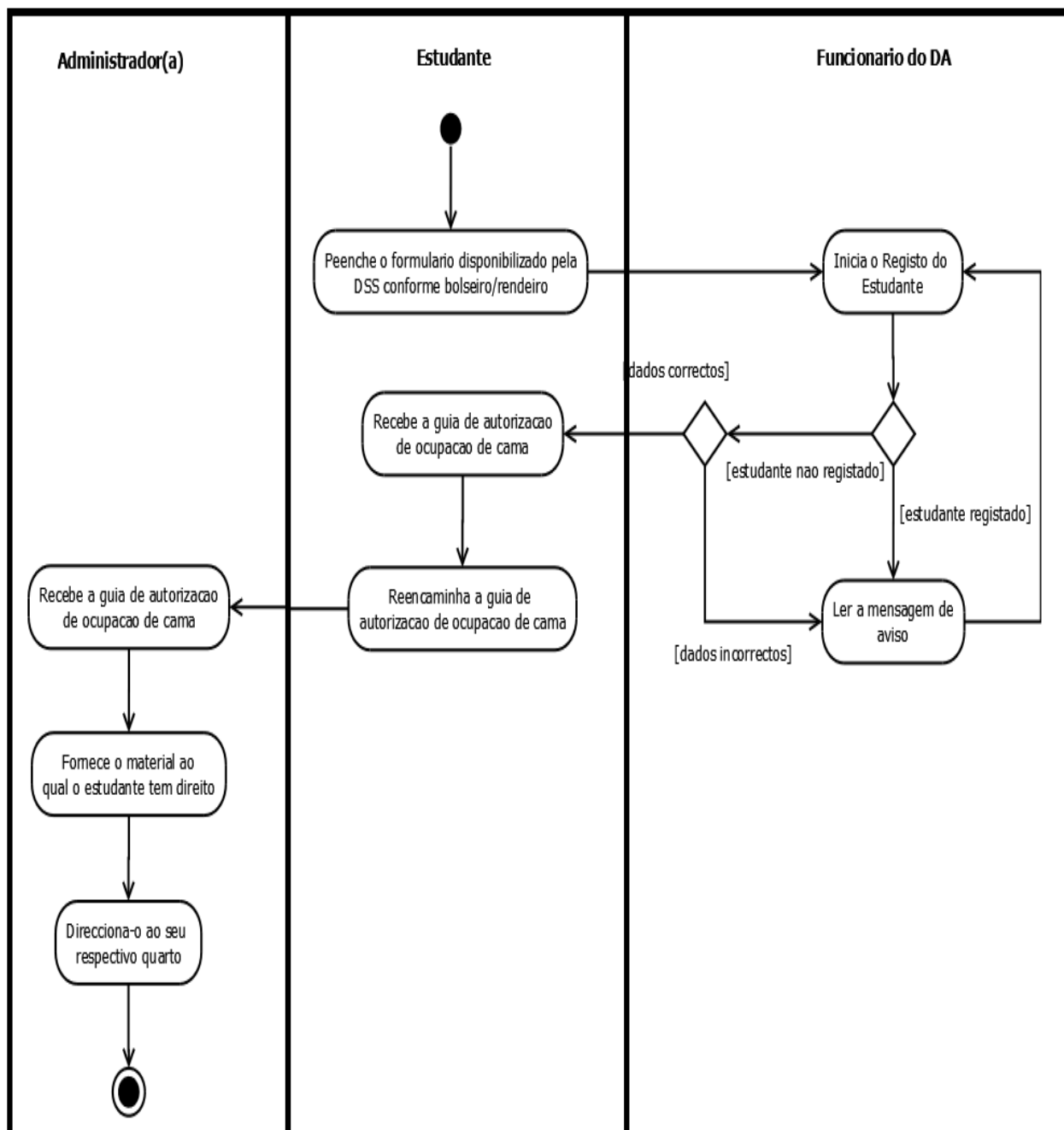


Figura 9 – Diagrama de Actividades: Registrar Estudante

### 4.7- Diagrama de Sequência

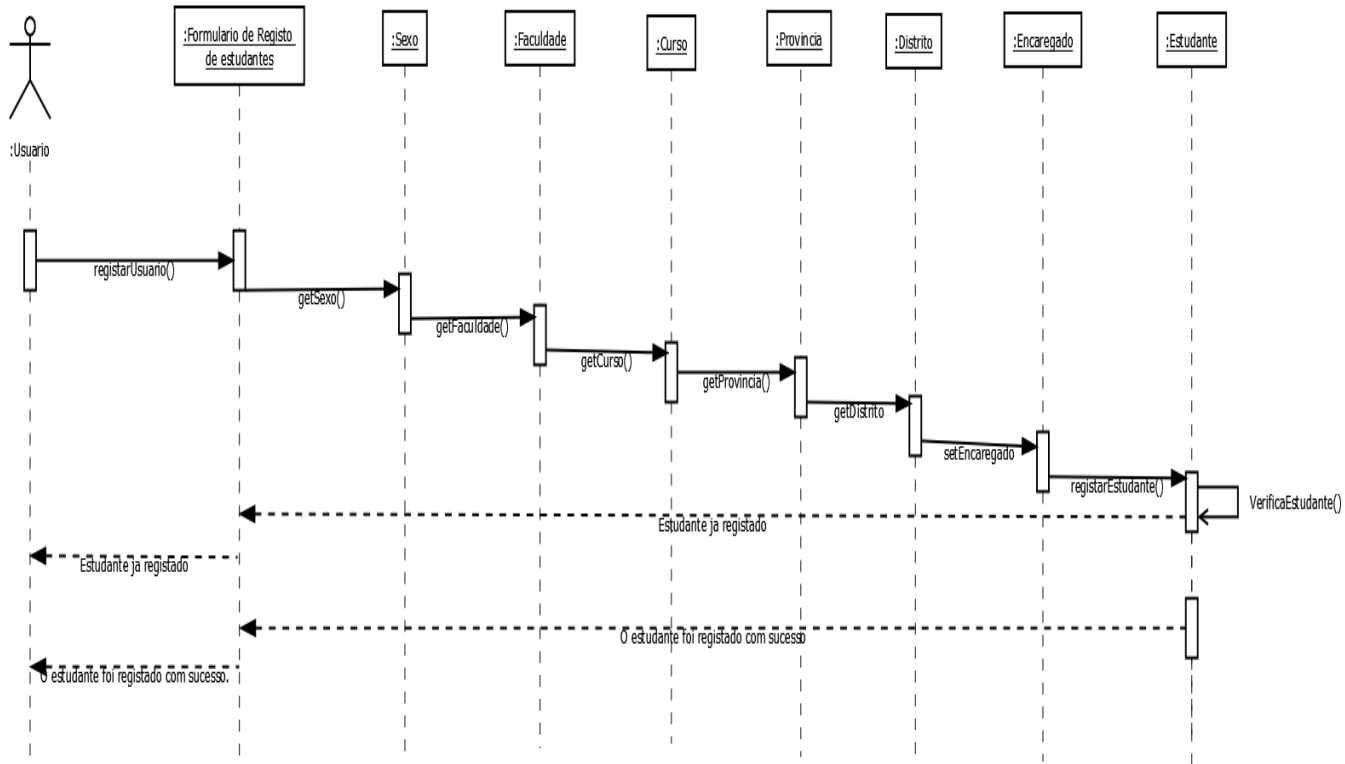


Figura 10 – Diagrama de Sequência: Registrar Estudante

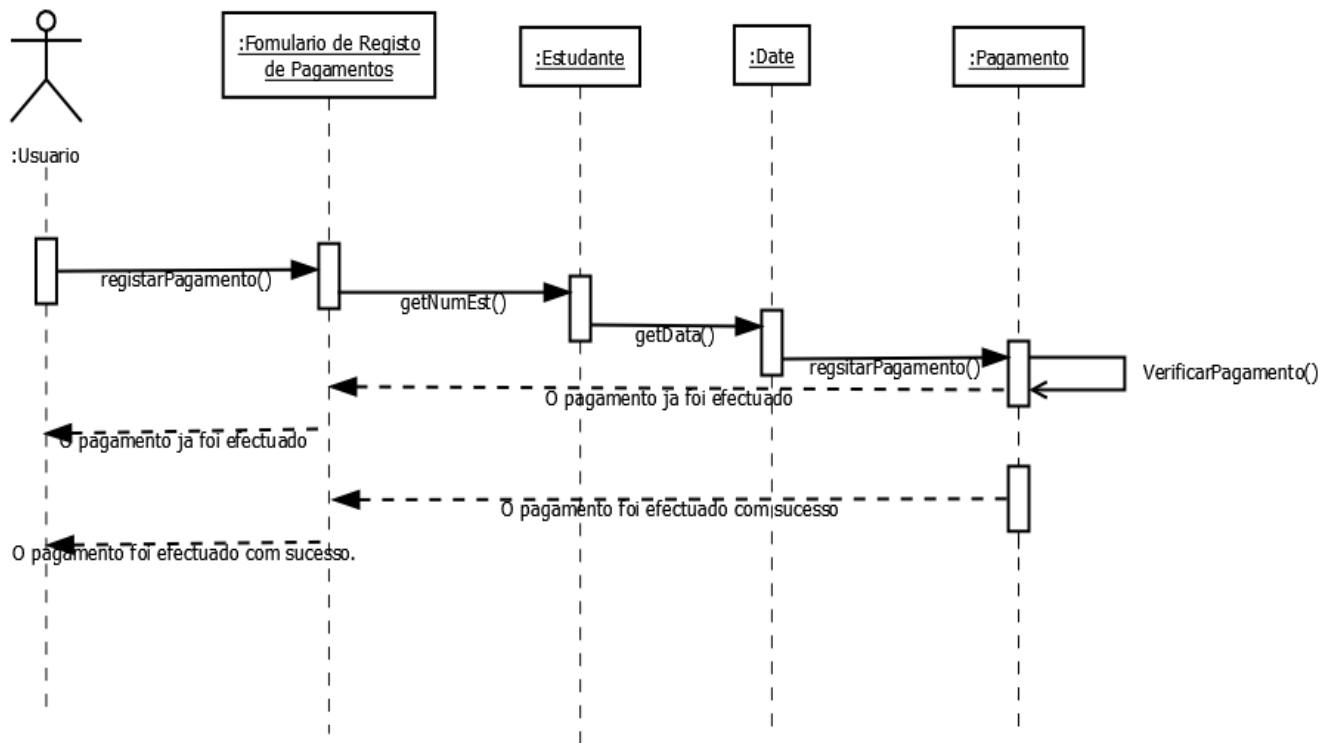


Figura 11 – Diagrama de Sequência: Registrar Pagamento

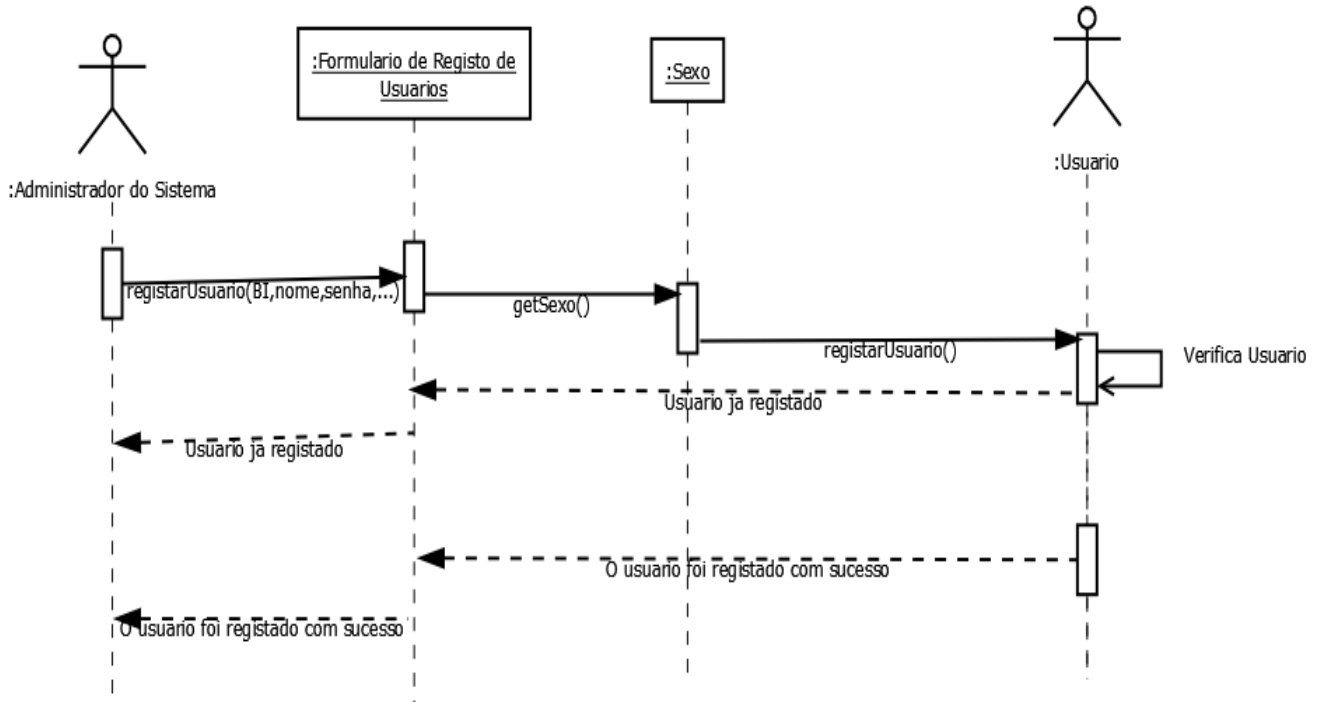


Figura 12 – Diagrama de Sequência: Registrar Usuário

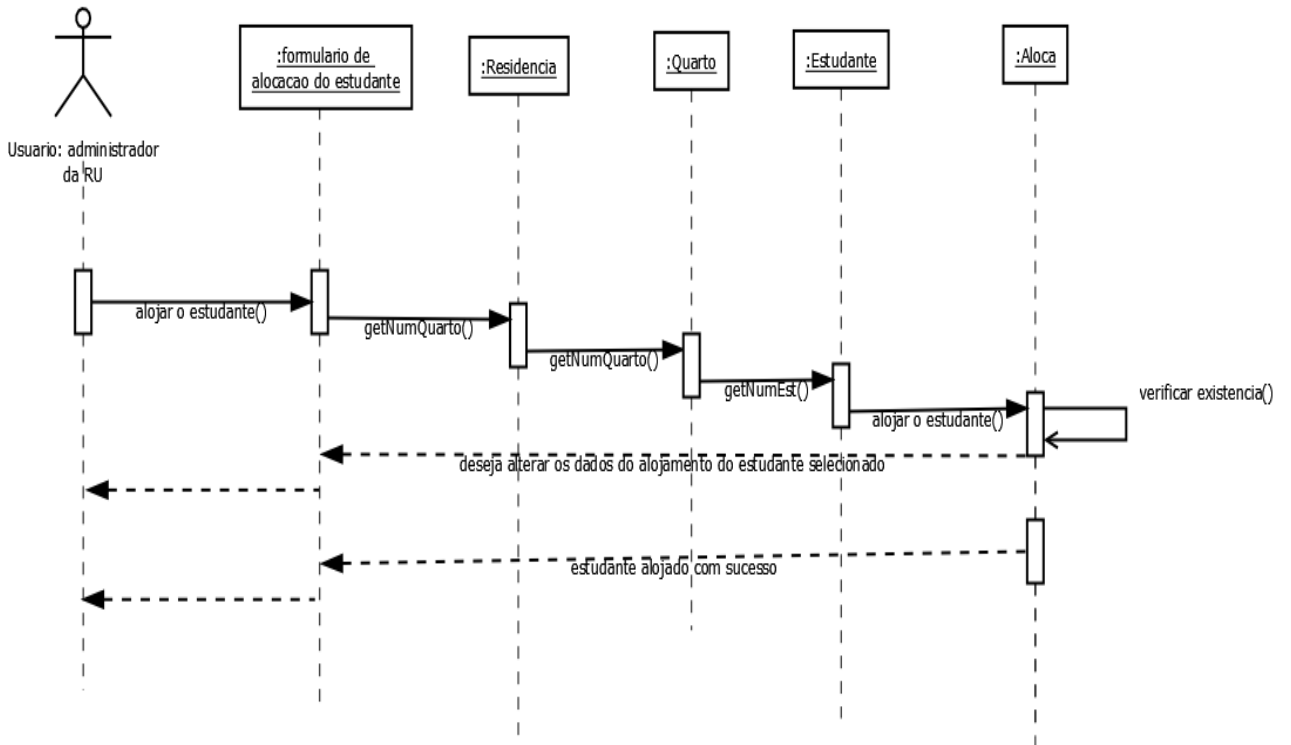


Figura 13 – Diagrama de Sequência: Alojamento do estudante

#### 4.7- Diagrama de Estados

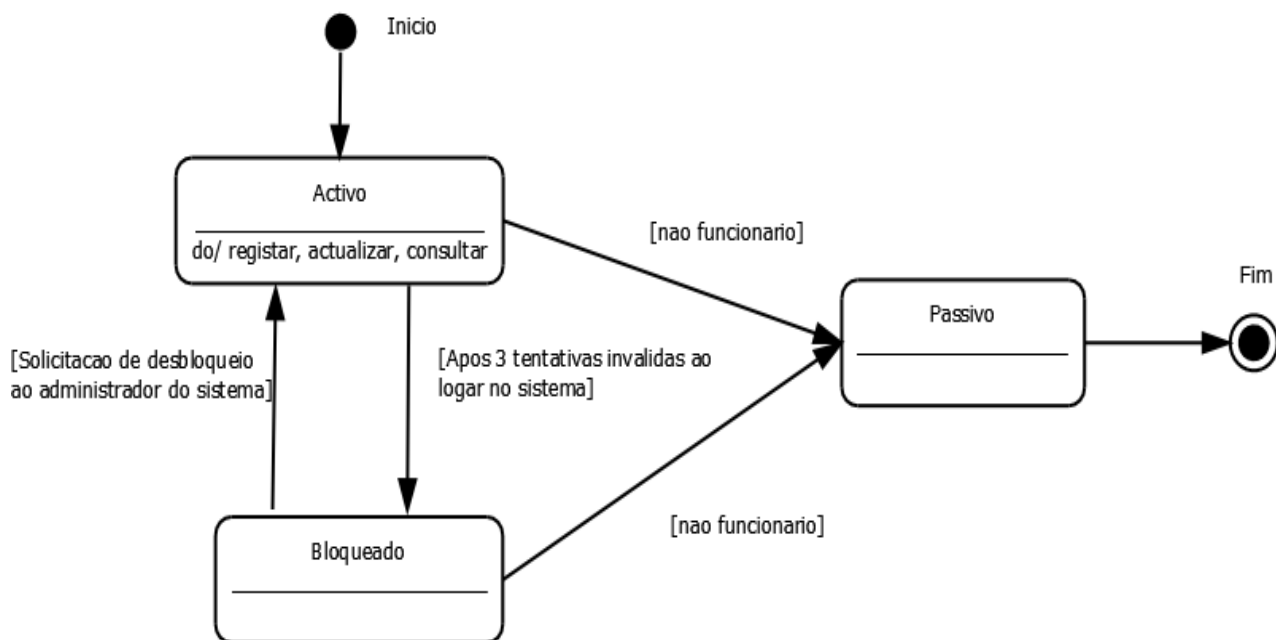


Figura 14 – Diagrama de Estado: Usuário

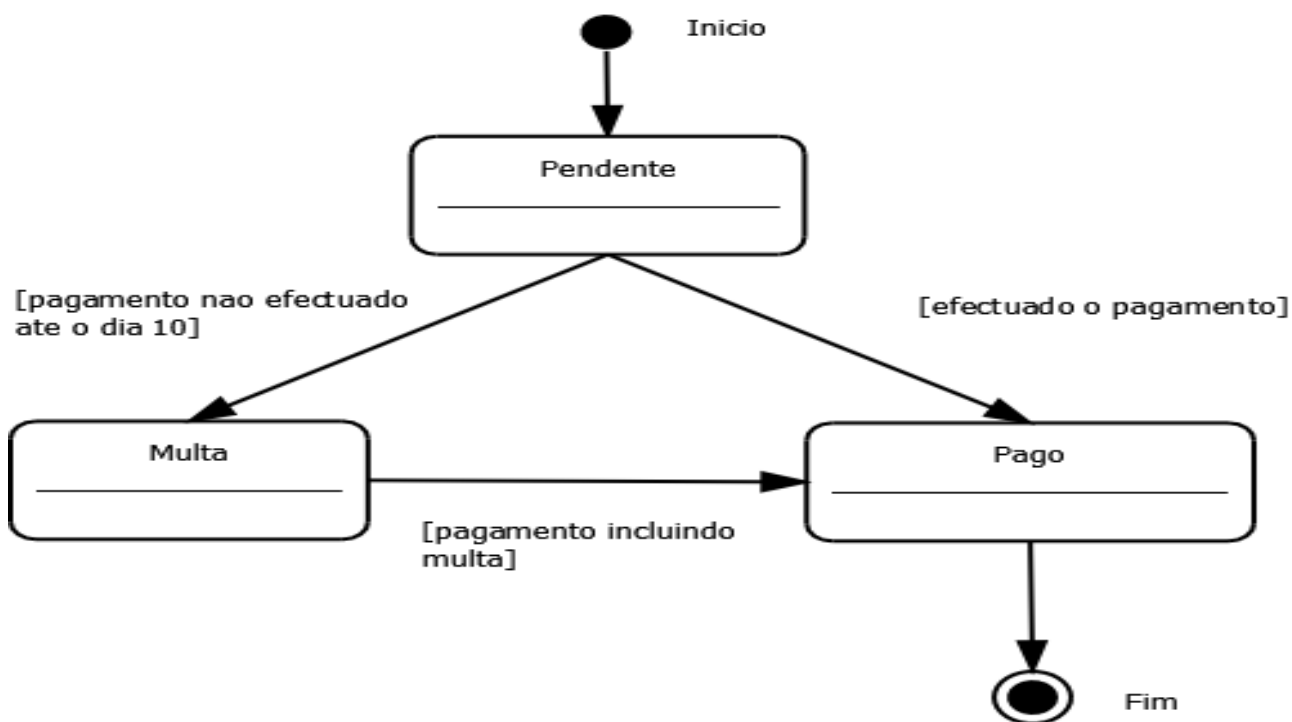


Figura 15 – Diagrama de Estado: Pagamento

## Capítulo VI – Conclusões e Recomendações

### 1- Conclusões

O modo de como flui a informação referente ao processo de alojamento e pagamentos de mensalidades implementado actualmente pela DSS, tem se mostrado ineficiente devido a quantidade de processos estudantis por eles administrados e a dispersão dos diversos funcionários da DSS que necessitam da mesma informação para poder tomar decisões.

Com a implementação do modelo proposto espera-se uma gestão melhorada, eficiente e racional dos recursos administrados pela DSS, visto que a informação será disponibilizada com maior facilidade através da *Web*, os utilizadores serão independentes da estação de trabalho, a manutenção passará a ser centralizada e a aplicação será independente da plataforma, uma vez que o único requisito é o navegador *Web* e a conexão à *Internet*.

Por outro lado, concluiu-se que não existe um modelo de processo de *software* adequado para o desenvolvimento de SI, uma vez que, a escolha do modelo de processo de *software* depende de diversos factores tais como: tempo, recursos económicos, recursos humanos, etc.

### 2- Recomendações

Por uma questão de comodidade para o estudante e para aumentar ainda mais a eficiência do modelo proposto, recomendo que nas próximas investigações relacionadas com o modelo descrito no presente trabalho, se implemente uma funcionalidade de pagamento via próprio sistema e que mecanismos de segurança sejam levados em consideração, dado que não serão mais necessárias as verificações mensais efectuadas pela DAF e deslocações constantes a procura do administrador da RU.

Que a infra-estrutura organizacional (processos, pessoas, hardware) seja reverificada de modo a albergar o sistema.

Por fim, que se estude a integração da solução proposta com os restantes sistemas actualmente em operação na UEM, como por exemplo, o sistema de registo académico.

## Bibliografia

### Referenciada

CONSELHO UNIVERSITÁRIO DA UEM, (2010). *Regulamento Pedagógico*. Sala do Conselho Universitário, p.7.

CHEN, H. e CHENG, R., (2007). *ZK Ajax Without JavaScript™ Framework*. Potix Corporation. Disponível em <http://www.4shared.com/document/bL4nSrFA/ApressZKAjaxwithouttheJavascrpt.htm>. Obtido em 20 de Dezembro de 2010

DEITEL, H. M. e DEITEL, P. J., (1999). *Java™ How to Program*. Third Edition. New Jersey: Prentice Hall.

JACOBSON, I., BOOCH, G. e RUMBAUGH, J., (1999). *Unified Software Development Process*. Addison Wesley.

MAMEDES, H. S., (2006). *Segurança Informática nas Organizações*. FCA - Editora de Informática. Lda.

NUNES, M. e O'NEILL, H., (2003). *Fundamental de UML*. 2ª Edição Actualizada e Aumentada. Lisboa: FCA – Editora de Informática. Lda.

PEREIRA, J. L. (1998). *Tecnologia de Base de Dados*. 3ª Edição. FCA - Editora de Informática. Lda.

PFLEEGER, S. L. e ATLEE, J. M., (2006). *Software Engineering: Theory and Practise*. Third Edition. Pearson Internation Edition.

PRESSMAN, R. S., (2006). *Engenharia de software*. 6ª Edição. São Paulo: MCGrawHill.

RODRIGUES, L. S., (2002). *Arquitecturas dos Sistemas de Informação*. FCA – Editora de Informática. Lda.

SOMMERVILLE, I., (2003). *Engenharia de Software*. 6ª Edição. São Paulo: Pearson Addison Wesley, p. 43-44.



SOMMERVILLE, I., (2007). *Engenharia de Software*. 8ª Edição. São Paulo: Pearson Addison Wesley.

WELLING, L. e THOMSON, L., (2001). *PHP e MySQL Web Development*. Indiana: Indianapolis.

### **Não Referenciada**

BLAHA, M. e PREMERLANI, W., (1998). *Object-Oriented Modelling for Database Application*. New Jersey: Prentice Hall.

BOAVENTURA, E. M., (2007). *Metodologia de Pesquisa*. São Paulo: Atlas.

WELLS, D. (1999). *Extreme Programming: A gentle Introduction*. Disponível em <http://www.extremeprogramming.org/>. Obtido em 16 de Dezembro de 2010

NATALI, A. C. *Pós Graduação em Engenharia de Software*. Disponível em [http://webx.sefaz.al.gov.br/posEngSoft/documentos/qos1/Pos\\_Maceio\\_Parte2\\_CicloVida.pdf](http://webx.sefaz.al.gov.br/posEngSoft/documentos/qos1/Pos_Maceio_Parte2_CicloVida.pdf). Obtido em 15 de Janeiro de 2011

RATIONAL SOFTWARE CORPORATION (1998). *Rational Unified Process: Best Practices for Software Development Teams*. Disponível em <http://www.dimap.ufrn.br/~jair/ES/artigos/rup.pdf>. Obtido em 14 de Fevereiro de 2011

SILVA, E. L. e MENEZES, E. M., (2001). *Metodologia da Pesquisa e Elaboração de Dissertação*. 3ª Edição Revisada e Atualizada. Florianópolis.

WHITEHEAD, P. e MARAN, R., (1997). *Internet and World Wide Web*. 2<sup>nd</sup> edition. IDG Books Worldwide .Inc

THE APACHE SOFTWARE FOUNDATION, (2010). *Apache Tomcat*. Disponível em <http://tomcat.apache.org>. Obtido em 1 de Dezembro de 2010

## Anexos

### Anexo 1: Guião de Entrevista



UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

#### GUIÃO DE ENTREVISTA

Este documento tem como objectivo recolher a informação referente ao processo de alojamento e pagamentos de mensalidades da DSS, uma vez que, nem tudo está documentado e existem aspectos não documentados que são relevantes para a pesquisa.

**Observação:** Denotar que, este documento não poderá ser usado para quaisquer outros fins senão para a elaboração do trabalho de final de curso.

Sendo assim, a seguir se apresentam algumas das questões que serão efectuadas durante as entrevistas:

#### Organização

1. Qual é o objectivo ou função da DSS?
2. Como é o organograma da DSS?
3. Como flui a informação referente ao estudantes e pagamentos de mensalidades na DSS?

#### Residências

1. Quantas residências a DSS possui e diga as respectivas localizações?
2. Qual é a quantidade de estudantes que vivem em cada residência?
3. Qual é a função dos administradores das RU?

4. Quando é que o DA envia a lista de estudantes e os novos processos estudantis aos administradores das RU?

### **Registo de Estudantes**

1. Actualmente, usa-se um sistema para registar os estudantes que vivem nas RU?
  - Caso exista, quais são os principais problemas encontrados no sistema de informação actual?
2. O estudante rendeiro recebe a guia de autorização de ocupação de cama, antes de ser alocado a um determinado quarto da RU?

### **Pagamentos de Mensalidades**

1. Quem regista e onde são registados os pagamentos de mensalidades?
2. Quem confirma e onde são confirmados os pagamentos de mensalidades?
3. Se o estudante pagar n meses adiantado, recebe um ou n recibos comprovativos de pagamento de mensalidades?

## Anexo 2: Ficha de Internamento



UNIVERSIDADE EDUARDO MONDLANE

DIRECÇÃO DOS SERVIÇOS SOCIAIS

DEPARTAMENTO DE ALOJAMENTO

### FICHA DE INTERNAMENTO

FICHA N° \_\_\_\_\_/\_\_\_\_

RESIDÊNCIA N° \_\_\_\_\_

QUARTO N° \_\_\_\_\_

ENTRADA \_\_\_/\_\_\_/\_\_\_

SAIDA \_\_\_/\_\_\_/\_\_\_

Nome do Estudante \_\_\_\_\_ filho  
de \_\_\_\_\_ e  
de \_\_\_\_\_ natural  
de \_\_\_\_\_ Província de \_\_\_\_\_, data de  
nascimento \_\_\_/\_\_\_/\_\_\_, cartão de estudante N° \_\_\_\_\_  
B. I. N° \_\_\_\_\_, emitido pelo arquivo de identificação  
de \_\_\_\_\_ em \_\_\_/\_\_\_/\_\_\_, n° de telemóvel  
pessoal \_\_\_\_\_ Proveniência \_\_\_\_\_

Nome do encarregado de educação \_\_\_\_\_, Grau de  
Parentesco \_\_\_\_\_ Profissão \_\_\_\_\_ local \_\_\_\_\_ de  
Trabalho \_\_\_\_\_ Endereço \_\_\_\_\_ :  
Av/Rua \_\_\_\_\_ Bairro \_\_\_\_\_

Localidade/Distrito/Província \_\_\_\_\_ Telefone/Telemóvel  
N° \_\_\_\_\_ Em caso de doença  
contactar \_\_\_\_\_ Av./Rua \_\_\_\_\_  
local de trabalho \_\_\_\_\_ Telefone/Telemóvel N° \_\_\_\_\_

Localidade/Distrito/Província \_\_\_\_\_ Data de ingresso na  
UEM \_\_\_\_/\_\_\_\_/\_\_\_\_ Curso \_\_\_\_\_  
Faculdade \_\_\_\_\_ Tipo de bolsa \_\_\_\_\_  
Nível \_\_\_\_ Tem problema de saúde? \_\_\_\_\_ Quais ? \_\_\_\_\_  
Vai passar refeições nos refeitórios da DSS? Todas \_\_\_\_ Algumas \_\_\_\_ Nenhumas \_\_\_\_  
Maputo, aos \_\_\_\_ de \_\_\_\_\_ de 20 \_\_\_\_

Conferido por

Assinatura do Estudante

### **Anexo 3: Contrato de arrendamento de cama**



**UNIVERSIDADE EDUARDO MONDLANE**

**DIRECÇÃO DOS SERVIÇOS SOCIAIS**

### **Termo de Contrato**

#### Concessão para aluguer de camas nas residências universitárias

#### **Tipo de contrato**

O contrato de arrendamento de camas é de carácter precário, isto é, a sua existência depende da disponibilidade de vagas nas residências universitárias.

O contrato é individual, isto é, o estudante paga para ele próprio ocupar uma cama e não para ocupá-la através de terceiros.

Enquanto não houver ordem para abandono da residência, os estudantes tem a obrigação de pagar as rendas dos meses em que consta como rendeiro, ocupando fisicamente ou não a respectiva cama.

Caso o estudante tenha terminado o contrato e não tenha devolvido as chaves do quarto será cobrado o valor da renda do respectivo mês.

### **Tipo de residência**

Serão alugadas camas nas residências nº 1, 2, 5, 6 e 8 para estudantes do sexo masculino, residências nº 7 para ambos os sexos e residência nº 4 para estudantes do sexo feminino.

### **Condições de pagamento**

O pagamento deve ser feito em depósito directo no Banco Standard Bank, na conta nº 106.004462.100.9, até dia 10 de cada mês, caso o ocupante não pague até à data estipulada, o valor será acrescido em 10% de multa; após o cumprimento deste prazo, e a renda não ter sido paga, o ocupante será expulso da residência, sem possibilidade de renovação do contrato.



**UNIVERSIDADE EDUARDO MONDLANE**

**DIRECÇÃO DOS SERVIÇOS SOCIAIS**

**Visto**

O Director

\_\_\_\_\_  
(dr. Arlindo Chongo)

### **Termo de Compromisso**

\_\_\_\_\_, filho de \_\_\_\_\_ e  
de \_\_\_\_\_ natural \_\_\_\_\_ de \_\_\_\_\_ nascido  
em \_\_\_/\_\_\_/\_\_\_ portador do BI N° \_\_\_\_\_ emitido em  
\_\_\_/\_\_\_/\_\_\_ pelo arquivo de identificação de \_\_\_\_\_ estudante  
matriculado no curso de \_\_\_\_\_ faculdade  
de \_\_\_\_\_ comprometo-me:

A cumprir e a fazer cumprir com zelo e dedicação o regulamento das residências estudantis e outras normas dele decorrentes.

A ceder a cama que temporariamente alugo, assim que o departamento precisar; a pagar as rendas regularmente e dentro dos prazos estabelecidos (até dia 10 de cada mês).

Maputo, aos \_\_\_\_\_ de \_\_\_\_\_ de 200 \_\_\_\_

O Estudante

O Chefe do Deptº de Aloj.

\_\_\_\_\_

\_\_\_\_\_

## **Vigência do Contrato**

O contrato é vigente a partir da data da sua assinatura, com validade até 21 de Dezembro do mesmo ano.

A renovação do contrato não é automática, carecendo de requerimento dirigido ao Director dos Serviços Sociais da U.E.M.

A Direcção dos Serviços Sociais só poderá renovar contratos com os estudantes rendeiros cujos seus pagamentos estão em dia, isto é, não tendo dívida na sua renda anterior.

## **Obrigação do ocupante**

O ocupante está obrigado a cumprir e a fazer cumprir com zelo e dedicação o regulamento das residências estudantis e outras normas deles decorrentes.

## **Sanções do ocupante**

O ocupante está sujeito as penalizações impostas pelo regulamento das residências universitárias, em caso de infracções do mesmo, uso indevido dos bens das residências

## **O valor da renda**

O valor estimado para pagamento mensal, por cada cama ocupada é de 750.00 MT (Setecentos e Cinquenta Meticais). Este valor pode sofrer alguma alteração mas com conhecimento prévio do reheiro.

Maputo, aos \_\_\_\_\_ de \_\_\_\_\_ de 20\_\_\_\_

O Estudante

O Director

\_\_\_\_\_

\_\_\_\_\_



## Anexo 4: Guia de Autorização de Ocupação de Cama



UNIVERSIDADE EDUARDO MONDLANE

DIRECÇÃO DOS SERVIÇOS SOCIAIS

DEPARTAMENTO DE ALOJAMENTO

### AUTORIZAÇÃO DE OCUPAÇÃO

Autorizo o(a) senhor(a) \_\_\_\_\_ de  
nacionalidade \_\_\_\_\_ da faculdade de \_\_\_\_\_

Curso de \_\_\_\_\_ para ocupar o quarto n° \_\_\_\_\_ da residência  
n° \_\_\_\_\_

Nota: V. Excia tem um prazo de 8 (oito) dias para ocupar o quarto indicado

Maputo, aos \_\_\_\_ de \_\_\_\_\_ de 200 \_\_\_\_

O Chefe do Dept° de Alojamento

\_\_\_\_\_

-----  
Informação da Residência n° \_\_\_\_\_

O Estudante \_\_\_\_\_ está alojado  
no quarto n° \_\_\_\_\_

Maputo, aos \_\_\_\_ de \_\_\_\_\_ de 200 \_\_\_\_

O (A) Administrador (a) da residência

\_\_\_\_\_

## **Anexo 5: Manual do Utilizador**

### **1- Introdução**

O presente manual foi desenvolvido com objectivo de dar apoio básico aos usuários que efectuarão as diversas operações (registo, actualização e outras) na aplicação *Web* de Gestão de Residências Universitárias, visto que os utilizadores podem enfrentar dificuldades ao tentar efectuar determinadas operações no sistema, por falta de domínio da aplicação ou por esquecimento do modo de operar com o sistema.

Sendo assim, o manual explica de forma breve, clara e objectiva os principais módulos do sistema (aplicação *Web* de Gestão de Residências Universitárias), de modo a facilitar os usuários a interagirem com o sistema.

Esta aplicação pode ser visualizada em vários navegadores como, por exemplo, Internet Explorer, Mozilla Firefox, Opera, Google Chrome, entre outros. Denotar que, caso o navegador utilizado para visualizar a aplicação *Web* não suporte a linguagem de estilo para apresentação de documentos HTML ou XML denominada *Cascade Style Sheet* (CSS), o sistema aparecerá com algumas diferenças na parte gráfica da aplicação.

No entanto, na elaboração deste manual foram usadas as seguintes convenções: os nomes dos campos e tabs são escritos a 'Itálicos', os nomes dos botões são escritos a boldado e entre aspas ("").

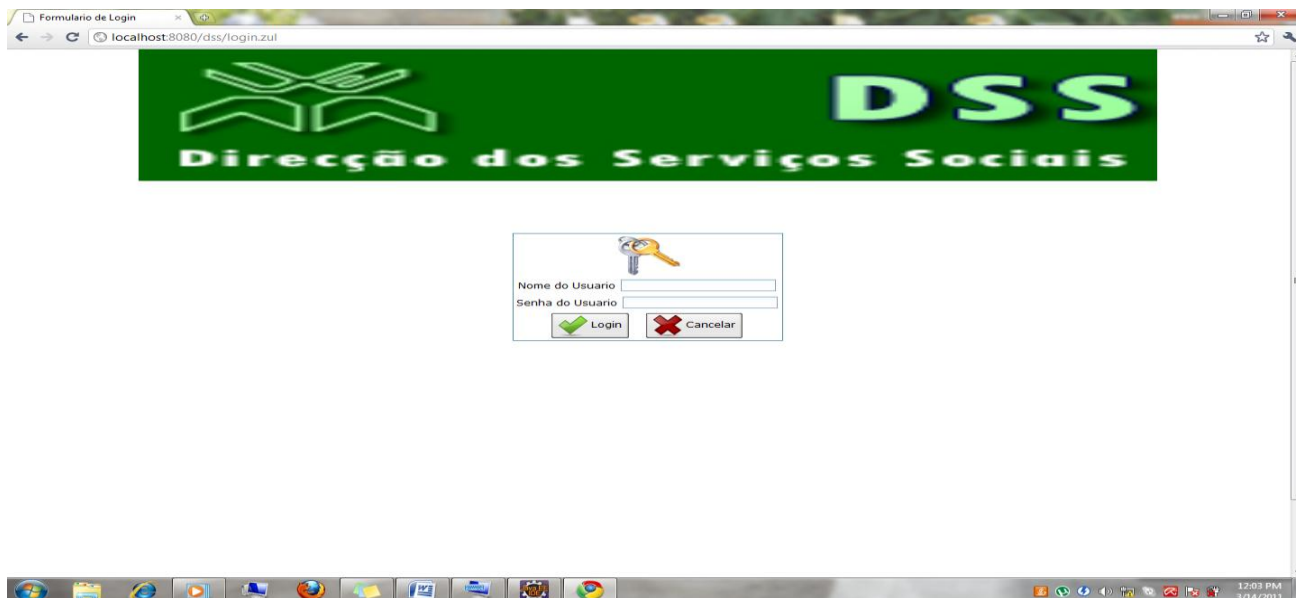
Para exploração da aplicação, o servidor deve estar ligado e devem ser seguidas as instruções das sessões subsequentes.

### **2- Página de Acesso**

#### **2.1- Login**

Sempre que os usuários tentarem aceder ao sistema, a página de acesso será exibida. Esta página fornece mecanismos de segurança que impede que os usuários sem autorização possam acessar o sistema. A Figura 1 mostra a página de acesso e pode ser acedida do seguinte modo:

- a) Abra o seu *browser* e digite na barra de endereço o seguinte url:  
<http://localhost:8080/DSS/login.zul>;
- b) Preencha os campos nome do usuário e senha com dados válidos;
- c) Pressione o botão “**Login**” para logar-se ao sistema.



**Figura 1 – Página de Acesso.**

Se pressionares o botão “**Cancelar**”, o texto dos campos da figura 1 serão limpados.

## 2.2- Logout

Para sair da sessão clique o botão “**Logout**” que aparece na página principal.

## 3- Página Principal

Depois de uma autenticação válida é lhe disponibilizada a página principal ou inicial que apresenta os seguintes menus: Parametrizações, Contabilidade, Relatórios, Administração, Ajuda.

### 3.1- Parametrizações

No menu Parametrizações existem dois menus que são: Gerais e Secundárias.

O menu Gerais: contém os principais formulários do sistema.

O menu Secundárias: contém os formulários secundários (auxiliares).

### **3.1.1 Menu Gerais**

Este menu contém os seguintes itens:

Alocar Estudantes: permite chamar o formulário que permite trocar o estudante de residência.

Cursos: chama o formulário de registo e actualização do curso.

Estudantes: permite chamar o formulário de registo e actualização do estudante.

Faculdades: chama o formulário de registo e actualização da faculdade.

Residências: permite chamar o formulário de registo e actualização da residência.

Permutar de Quartos: chama o formulário que permite que o estudante permute de quarto.

Tipo de Bolsa: permite chamar o formulário de registo e actualização do tipo de bolsa.

Trocar o Estudante de Residência: chama o formulário que permite trocar o estudante de residência quando ele já está alocado a um determinado quarto.

### **3.1.2 Menu Secundárias**

Este menu possui os seguintes itens:

Andares: chama o formulário de registo e actualização dos andares de uma determinada residência.

Banco: chama o formulário de registo e actualização do banco.

Distritos: chama o formulário de registo e actualização do distrito.

Sexo: chama o formulário de registo e actualização do sexo.

Grau Parentesco: chama o formulário de registo e actualização do grau de parentesco.

Províncias: chama o formulário de registo e actualização da província.

Quartos: chama o formulário de registo e actualização do quarto de uma determinada residência.

### **3.2- Contabilidade**

Este menu contém os seguintes itens:

Pagamentos: chama o formulário de registo e actualização do pagamento referente ao arrendamento de cama nas RU.

### **3.3- Relatórios**

Contém vários menus, que disponibilizam os diversos tipos de relatórios sobre as residências universitárias.

### **3.4- Administração**

Possui o menu Gestão de Utilizadores, e este por sua vez possui o sub menu Criar Contas do Usuário que permite chamar o formulário de registo e actualização da conta do usuário.

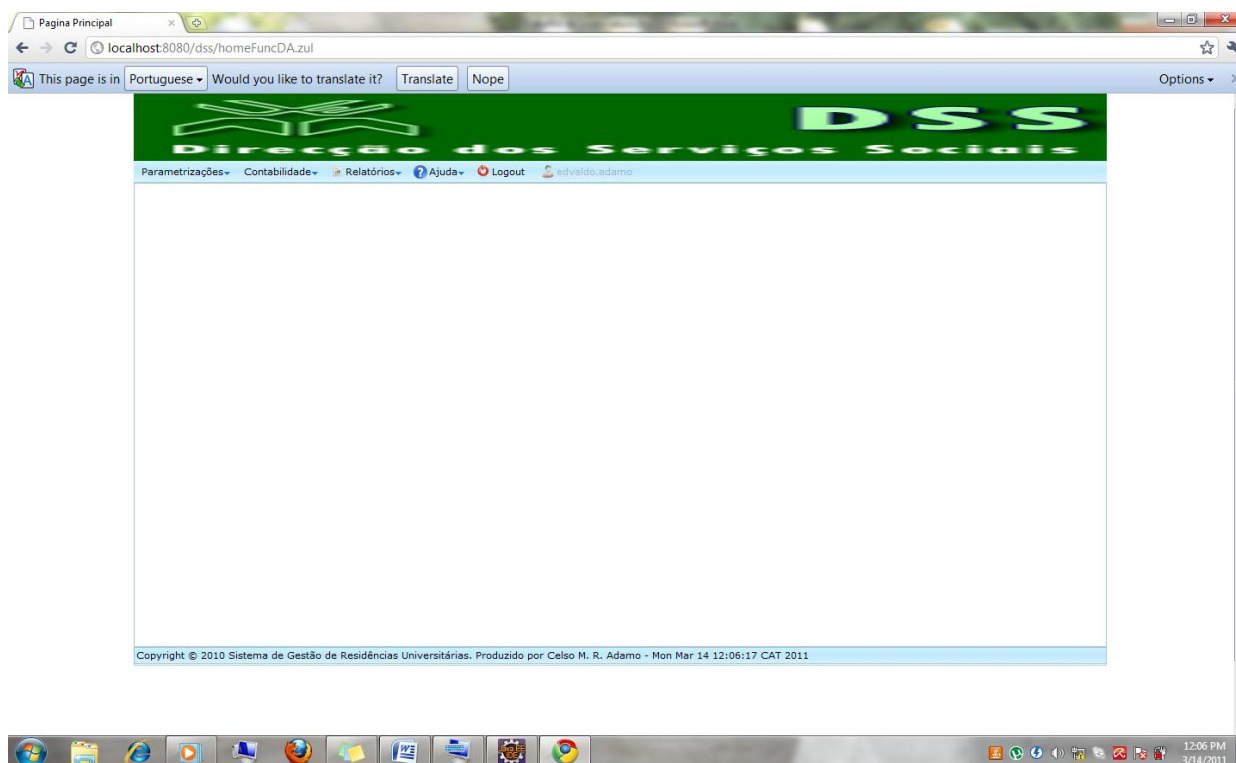
### **3.5- Ajuda**

Neste menu existem os seguintes menus: Sobre a aplicação e Sobre o autor.

Sobre a aplicação: permite chamar a janela que mostra detalhes referentes a aplicação.

Sobre o autor: permite chamar a janela que mostra os detalhes do autor da aplicação.

## Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM



**Figura 2 – Página Principal (usuário que pertence a categoria funcionário do DA)**

### NOTA:

- a) Os menus são disponibilizados de acordo a sua categoria. Cada categoria tem seus os privilégios, isto é, o (s) administrador (es) do sistema são os únicos com acesso ao menu administração, o funcionário da DAF só têm acesso ao menu contabilidade e relatórios que dizem respeito ao pagamento de mensalidades, etc.

#### 4- Registrar ou Actualizar um Usuário

Este formulário é acessado clicando em Administração → Gestão de Utilizadores → Criar Contas de Usuários e tem como finalidade registrar ou actualizar novos usuários no sistema. Neste formulário existem duas tabs, uma para registo do usuário (Figura 3) e outra para actualização do usuário (Figura 4). Por defeito o sistema aparece com a *tab de registo* já seleccionada. Nela preenche-se os campos em branco com a informação do usuário:

*Número de BI:* introduz-se o número do BI do usuário (funcionário);

*Nome do Funcionário:* introduz-se o nome do funcionário;

*Categoria:* é a categoria do funcionário, que pode ser: administrador do sistema, administrador da RU, funcionário do DA, funcionário da DAF, director;

*Sexo:* como o próprio nome diz, coloca-se o género do usuário;

*Nome do Usuário:* o nome com que o usuário se loga ao sistema e deve ser único;

*E-mail do Usuário:* este campo não é obrigatório e é usado pelo administrador do sistema para fornecer informações importantes ao usuário como, por exemplo, fornecimento da senha caso este tenha esquecido;

*Senha:* coloca-se a senha do usuário e esta deve ter no mínimo 6 caracteres;

*Confirmação da Senha:* deve confirmar a senha digitada acima;

*Estado:* define o estado do usuário.

- a) Se **activo**, o usuário pode logar-se ao sistema e efectuar as operações de direito;
- b) Caso contrário (**passivo**) não pode logar-se ao sistema e conseqüentemente não conseguirá efectuar nenhuma operação no sistema.

Para o administrador completar esta operação, deve clicar o botão “**Registrar**”.

The screenshot shows a web browser window displaying the 'Dados do Usuário' (User Data) form. The form is part of the 'DSS' (Direção dos Serviços Sociais) system. The form fields are as follows:

Field	Value
Numero do BI	050026103E
Nome do Funcionário	Carlos da Silva
Sexo	Masculino
Categoria	Director
Nome do Usuário	carlos.silva
E-mail do Usuário	carlos.silva@gmail.com
Senha	*****
Confirmação da Senha	*****
Estado	Activo

At the bottom of the form, there are two buttons: 'Registar' (with a green checkmark icon) and 'Cancelar' (with a red X icon). The browser's address bar shows 'localhost:8080/dss/homeAdminSist.zul'. The system's footer text reads: 'Copyright © 2010 Sistema de Gestão de Residências Universitárias. Produzido por Celso M. R. Adamo - Mon Mar 14 12:23:53 CAT 2011'.

**Figura 3 – Registo do Usuário.**

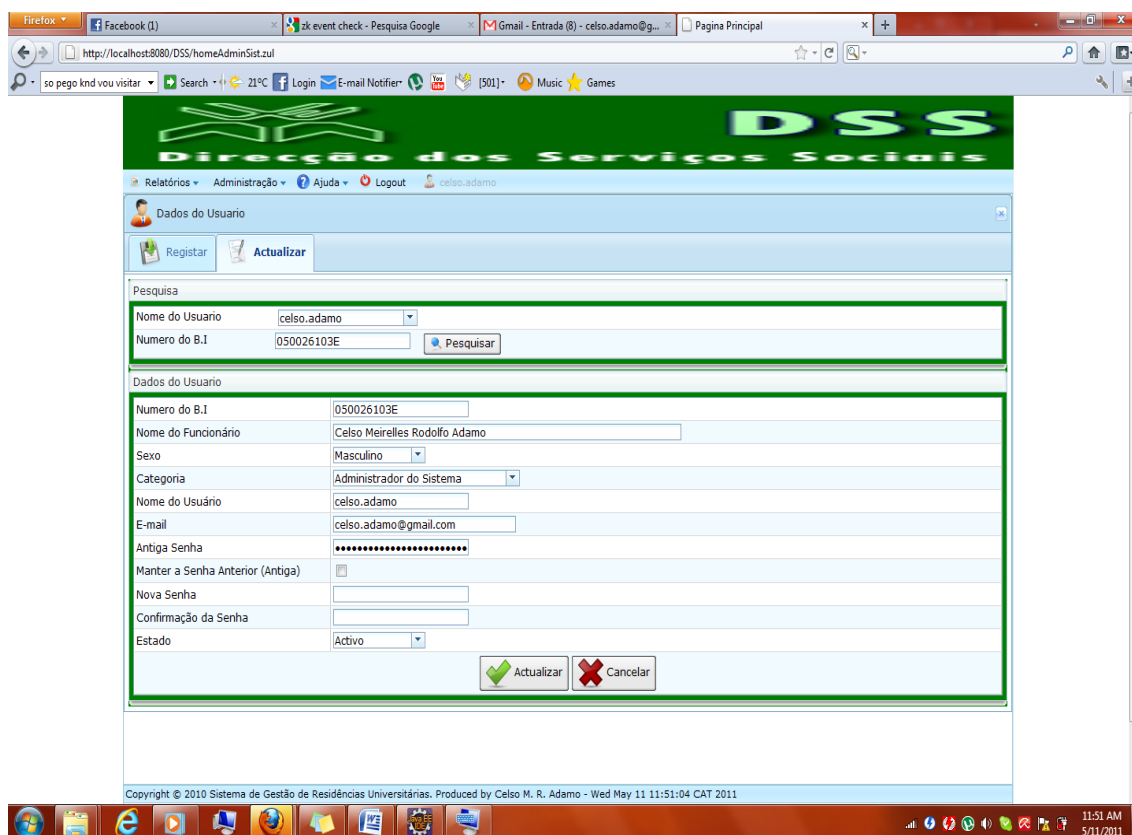
### **Registo bem sucedido**

Se todos os campos foram preenchidos correctamente, ao clicarmos o botão **“Registar”**, o sistema registará a conta do usuário e mostrará uma mensagem informativa referente a operação realizada com sucesso.

O botão **“Cancelar”** serve para limpar a informação de todos os campos do respectivo formulário.

Se clicares a *tab* **Actualização**, toda a informação preenchida anteriormente na *tab* **registo** pode ser actualizada. Para realizar esta operação, primeiramente deve-se seleccionar o nome do usuário que pretendes actualizar. De seguida aparecerá o número do BI do usuário seleccionado. Feito isso, deve-se clicar o botão **“Pesquisar”**. Após a pesquisa todos os campos do formulário são preenchidos. Finalmente, para completar esta operação, deve-se alterar os campos que assim o desejares e clicar no botão **“Actualizar”**.





**Figura 4 – Actualização do Usuário**

### **Actualização bem sucedida**

Se todos os campos foram preenchidos correctamente, ao pressionarmos o botão “**Actualizar**”, o sistema actualizará a conta do usuário e mostrará uma mensagem informativa referente a operação realizada com sucesso.

### **NOTA:**

- a) Todos os formulários contêm os botões “**Cancelar**” e tem sempre a mesma função de acordo com o grupo (formulário ou relatório).
- b) Sempre que o registo ou actualização seja efectuado com sucesso, aparece uma mensagem informativa de registo ou actualização bem sucedida respectivamente conforme o registo ou actualização.
- c) Cada formulário apresenta sempre duas abas, uma de registo e outra de actualização, excepto os formulários de escolha de parâmetros para geração dos diversos relatórios.

## 5- Registrar ou Actualizar Pagamentos

Este formulário é acessado clicando em Contabilidade → Pagamentos e tem como finalidade registrar ou actualizar os pagamentos no sistema. Nele existem duas tabs, uma para registo do pagamento (Figura 5) e outra para actualização da mesma (Figura 6). Como abordado acima, por defeito o sistema aparece com a *tab de registo* já seleccionada. Sendo assim, para o usuário poder registrar uma determinada residência, deve preencher correctamente o formulário de registo de pagamento e posteriormente pressionar o botão “Registrar”.

Copyright © 2010 Sistema de Gestão de Residências Universitárias. Produced by Celso M. R. Adamo - Sun Apr 03 21:02:17 CAT 2011

Figura 5 – Registo do Pagamento.

Ao clicares na *tab Actualização*, toda a informação preenchida anteriormente na *tab registo* pode ser actualizada. Para tal, primeiramente deve-se introduzir o número do estudante e clicar no botão “Pesquisar”. De seguida serão mostrados os dados referentes ao pagamento do estudante. Para terminar esta operação deve-se preencher correctamente os campos ilustrados na figura abaixo e clicar no botão “Actualizar”.

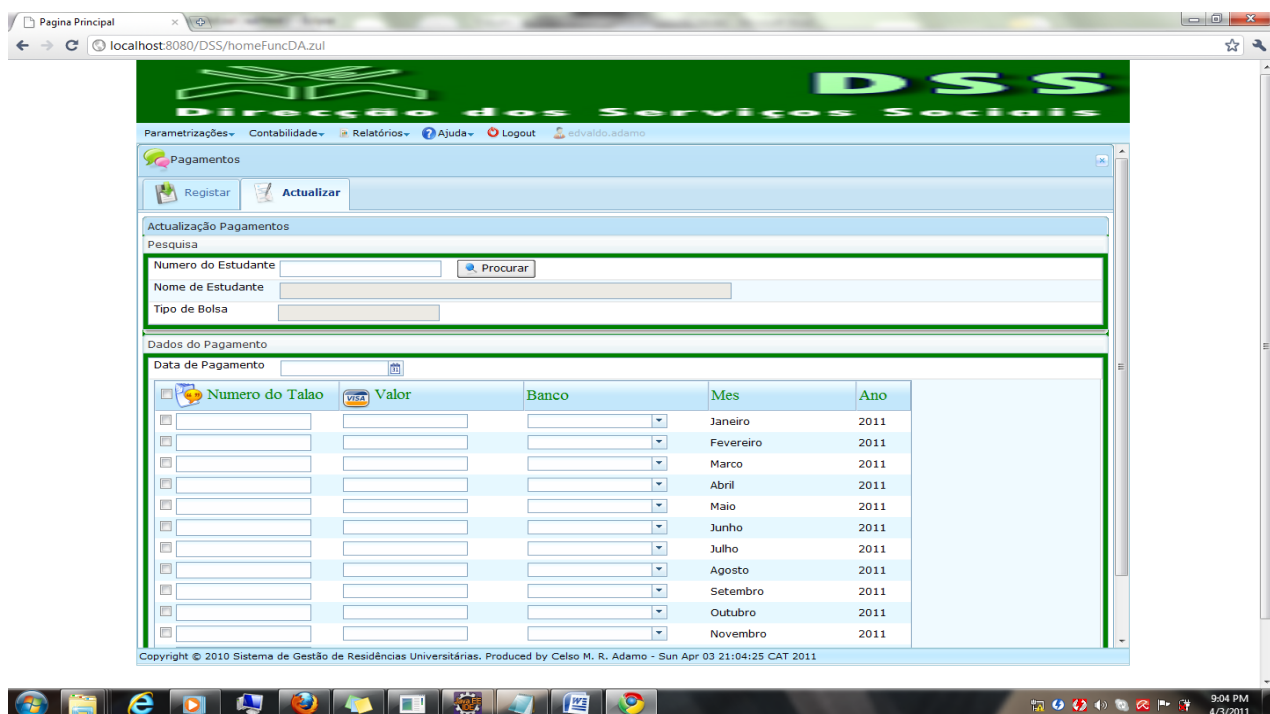


Figura 6 – Actualização do Pagamento

## 6- Registrar ou Actualizar um Estudante

Este formulário é acessado clicando em Parametrizações→Gerais→Estudante e tem como finalidade registrar ou actualizar dados do estudante no sistema. O mesmo (formulário) possui duas tabs, uma para registo dos dados do estudante e outra para actualização dos mesmos. O formulário da *tab de registo* divide-se em 3 secções (dados do estudante, dados de ingresso e dados do encarregado de educação). Portanto, para efectuar o registo do estudante, deve-se preencher correctamente todos os campos obrigatórios conforme a figura abaixo e de seguida clicar no botão “**Registrar**”.

## Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM

The screenshot shows a web browser window displaying the 'Direção dos Serviços Sociais' (DSS) system. The page title is 'Estudante' and it features a navigation menu with options like 'Parametizações', 'Contabilidade', 'Relatórios', 'Ajuda', and 'Logout'. Below the menu, there are buttons for 'Registrar' and 'Actualizar'. The main section is titled 'Dados do Estudante' and contains a form with the following fields:

Numero de Estudante	20072041007
Numero do BI	050026103E
Nome do Estudante	Francisco Alvaro Marta Adamo
Data de Nascimento	2011-03-07
Sexo	Masculino
Provincia de	Tete
Natural de	Moatize
Nacionalidade	Mocambicana
Filho de	Francisco Alvaro Marta Adamo
E de	Angela Maria Lucas Rodolfo Adamo
Morada	Karil Marx

At the bottom of the form, there is a section for 'Ingresso a UEM' and a copyright notice: 'Copyright © 2010 Sistema de Gestão de Residências Universitárias. Produced by Celso M. R. Adamo - Mon Mar 14 19:11:45 CAT 2011'.

**Figura 7 – Registo dos dados do estudante.**

Ao clicares na *tab* *Actualização*, primeiramente digita-se o número do estudante e de acordo com este número todos os campos são completados com os dados do estudante. Feito isto, deve-se preencher correctamente e conforme o desejado, os campos ilustrados na figura 8 e clicar no botão “**Actualizar**”.

The screenshot shows the same web browser window, but now displaying the 'Actualização dos dados do estudante' form. The page title is 'Estudante' and it features a navigation menu with options like 'Parametizações', 'Contabilidade', 'Relatórios', 'Ajuda', and 'Logout'. Below the menu, there are buttons for 'Registrar' and 'Actualizar'. The main section is titled 'Pesquisa' and contains a search form with the following fields:

Numero do Estudante	20072041007	Procurar
Nome do Estudante	Celso Meirelles Rodolfo Adamo	

Below the search form, there is a section titled 'Dados do Estudante' and a form with the following fields:

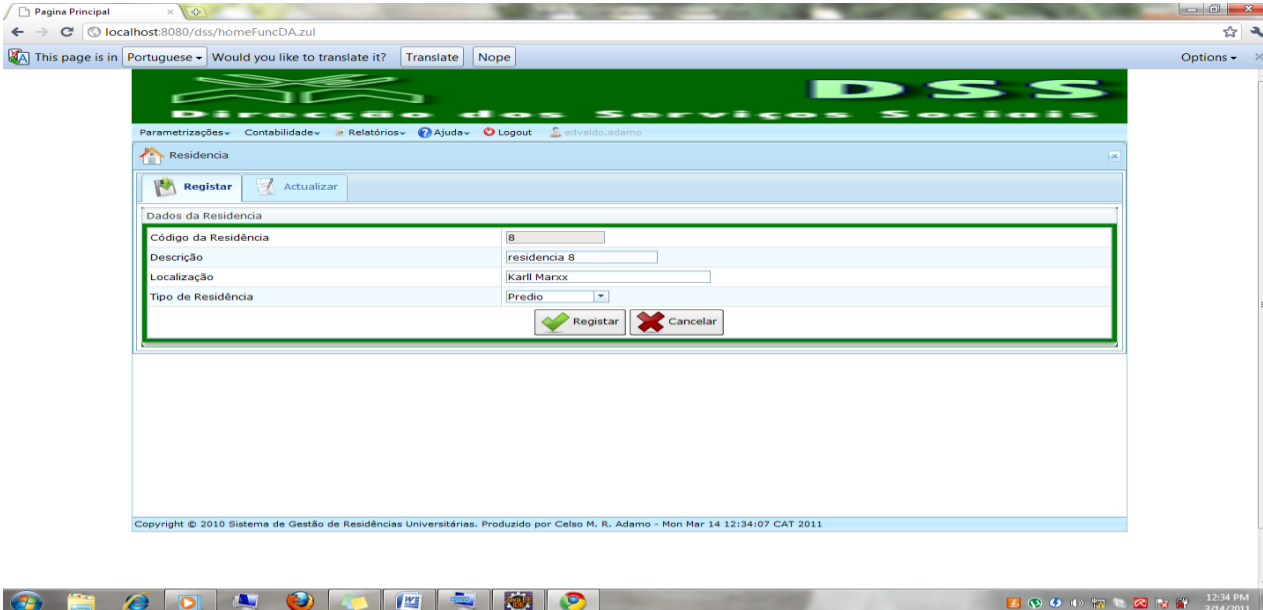
Nome do Estudante	Celso Meirelles Rodolfo Adamo
Numero de BI	0500261003E
Data de Nascimento	1989-05-06
Sexo	Masculino
Provincia de	Tete
Natural de	Moatize
Nacionalidade	Mocambicana
Filho de	Francisco Alvaro Marta Adamo
E de	Angela Maria Lucas Rodolfo Adamo

At the bottom of the form, there is a section for 'Ingresso a UEM' and a copyright notice: 'Copyright © 2010 Sistema de Gestão de Residências Universitárias. Produced by Celso M. R. Adamo - Mon Mar 14 19:12:27 CAT 2011'.

**Figura 8 – Actualização dos dados do estudante**

## 7- Registrar ou Actualizar uma Residência

Este formulário é acessado clicando em Parametrizações→Gerais→Residências e tem como finalidade registrar ou actualizar residências no sistema. Conforme a estrutura dos formulários acima, aqui também temos duas tabs: uma de registo e outra de actualização. Por defeito o sistema aparece com a *tab de registo* já seleccionada. Sendo assim, para o usuário poder registrar uma determinada residência, este deve preencher correctamente os campos ilustrados na figura 9 e clicar o botão “**Registrar**”.



The screenshot shows a web browser window with the URL localhost:8080/dss/homeFuncDA.zul. The page title is 'Pagina Principal'. The main content area is titled 'Residencia' and has two tabs: 'Registrar' (selected) and 'Actualizar'. The 'Registrar' tab contains a form with the following fields:

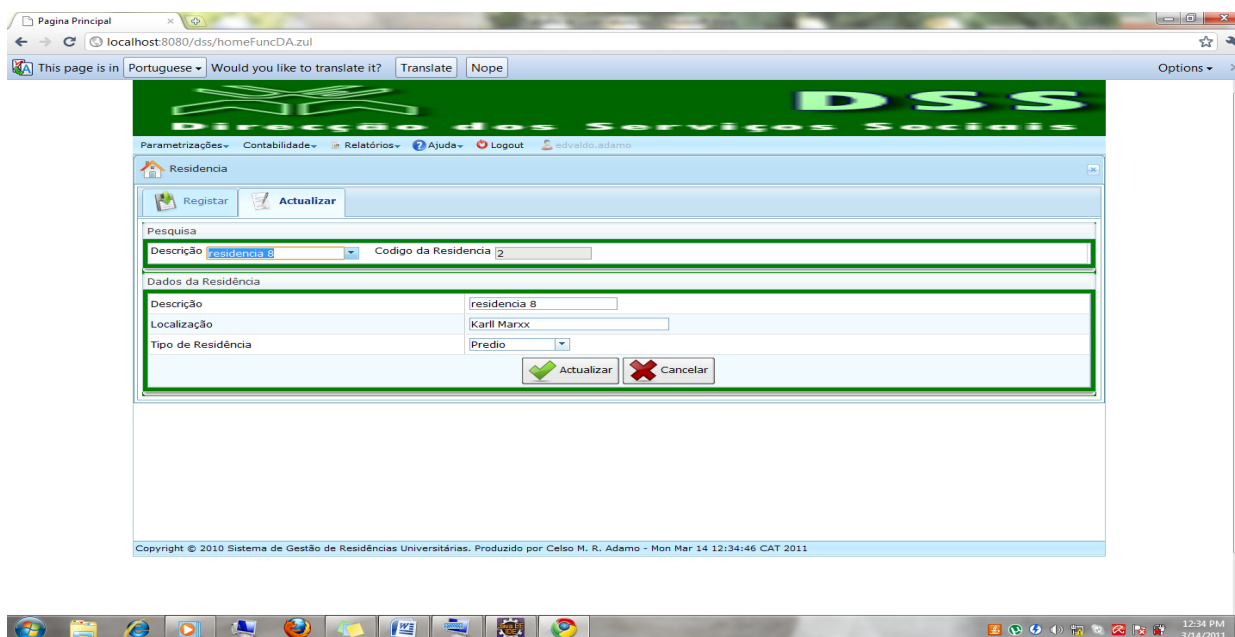
Dados da Residência	
Código da Residência	8
Descrição	residencia 8
Localização	Karl Marx
Tipo de Residência	Predio

At the bottom of the form are two buttons: 'Registrar' (with a green checkmark icon) and 'Cancelar' (with a red X icon). The footer of the page reads: 'Copyright © 2010 Sistema de Gestão de Residências Universitárias. Produzido por Celso M. R. Adamo - Mon Mar 14 12:34:07 CAT 2011'.

**Figura 9- Registo da Residência.**

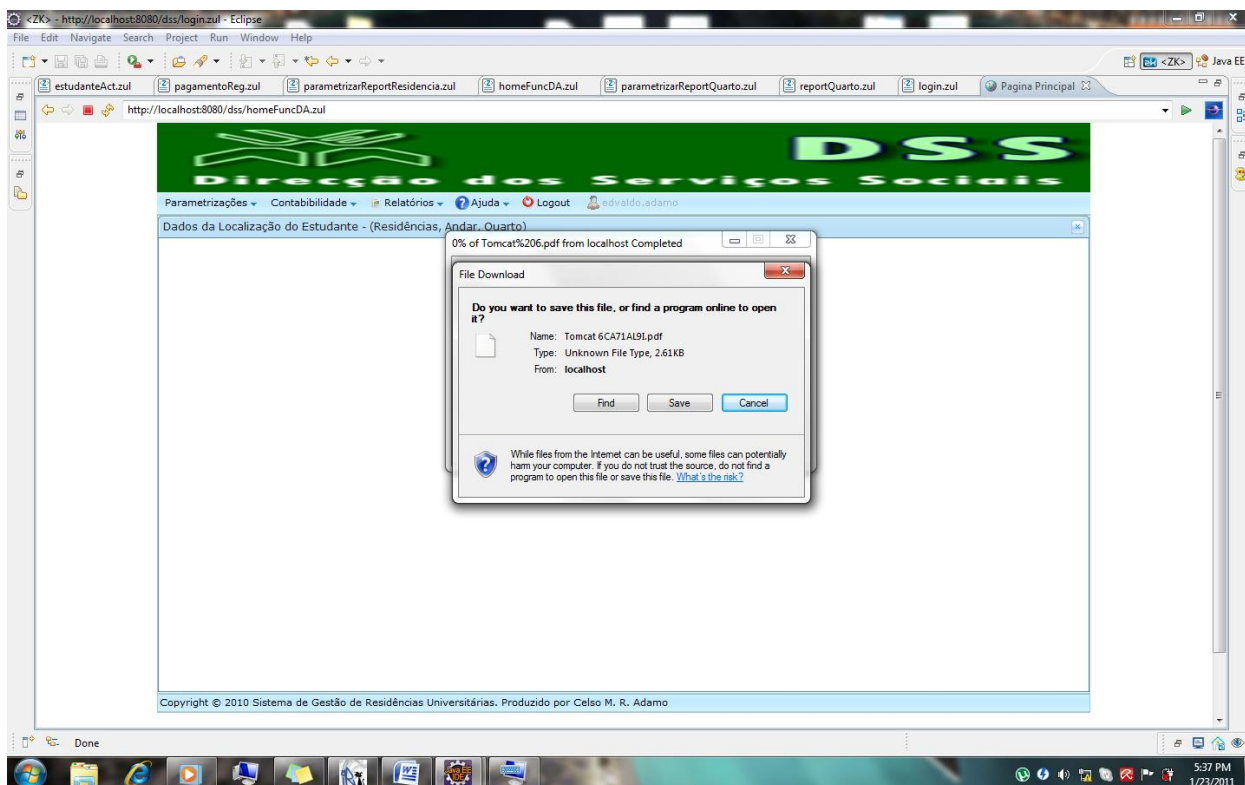
Ao clicares na *tab Actualização*, toda a informação preenchida anteriormente na *tab registo* pode ser actualizada, mas para tal, primeiramente deve-se seleccionar a residência que queiras actualizar. Após a selecção, todos os campos do formulário serão preenchidos com os dados anteriormente registados. Para finalizar deve-se preencher correctamente os campos ilustrados na figura abaixo e clicar no botão “**Actualizar**”.

## Sistema de Gestão de Alojamentos e Pagamentos de Mensalidades para Residências Universitárias da UEM



**Figura 12 – Atualização da Residência**

Por fim dizer que, se o seu computador não tiver instalado o programa para ler o formato (PDF, DOC, RTF, etc.) seleccionado para geração dos relatórios aparece a figura abaixo questionando se deseja salvar o arquivo ou encontrar um programa online para abri-lo.



**Figura 13 – Computador sem nenhum programa instalado que consiga ler o formato seleccionado**