

DEPARTAMENTO DE ENGENHARIA E TECNOLOGIA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

**CONCEPÇÃO E IMPLEMENTAÇÃO DE UM SOFTWARE PARA
GESTÃO DE APARTAMENTOS DA EMPRESA XBXB SEGUNDO
COMÉRCIO E PRESTAÇÃO DE SERVIÇO (SU), LDA, 2023.**

Autor: Lucas Falau Cundende

Orientador: Prof. Edivaldo de Carvalho, Lic.

Luanda, 2024

DEPARTAMENTO DE ENGENHARIA E TECNOLOGIA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

**CONCEPÇÃO E IMPLEMENTAÇÃO DE UM SOFTWARE PARA
GESTÃO DE APARTAMENTOS DA EMPRESA XBXB SEGUNDO
COMÉRCIO E PRESTAÇÃO DE SERVIÇO (SU), LDA, 2023.**

Trabalho do fim de curso apresentado ao Instituto Superior de Angola, Luanda, como requisito para obtenção do grau de Licenciatura em Engenharia Informática. sob orientação do Prof. Edivaldo de Carvalho, Lic.

Autor: Lucas Falau Cundende

Orientador: Prof. Edivaldo de Carvalho, Lic.

Luanda, 2024

DEPARTAMENTO DE ENGENHARIA E TECNOLOGIA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

**CONCEPÇÃO E IMPLEMENTAÇÃO DE UM SOFTWARE PARA
GESTÃO DE APARTAMENTOS DA EMPRESA XBXB SEGUNDO
COMÉRCIO E PRESTAÇÃO DE SERVIÇO (SU), LDA, 2023.**

O JÚRI

Presidente _____

1º Vogal _____

2º Vogal _____

Secretário _____

CLASSIFICAÇÃO: _____

DATA DA DEFESA _____ / _____ / _____

DEDICATÓRIA

Dedico este trabalho de Conclusão de Curso à minha amada Mãe Noémia Munge Falau, fonte inesgotável de amor, apoio e inspiração. Seu exemplo de dedicação e força sempre iluminou meu caminh, tornando este feito possível. Agradeço por ser a minha maior incentivadora e por ser a razão pela qual me esforço continuamente. Este trabalho é dedicado a você, minha eterna fonte de motivação.

AGRADECIMENTOS

A Deus por tudo que tem feito na minha vida, por me dar força para que eu pudesse chegar até aqui, e por estar comigo em todas as minhas caminhadas.

A minha Mãe Noémia Falau, à quem sou grato eternamente, por tudo que fez e tem feito para mim poder alcançar os meus objetivos, trabalhando arduamente para que nada me falte, e ser a melhor pessoa da minha vida.

Ao meu tio Teófilo Falau e a minha Avô Clementina Falau, pelo incentivo, apoio em todos os sentidos para que pudesse concluir a minha formação.

As minhas Mães, Isabel Falau, Celestina Falau e Leonora Falau, por serem as pessoas que me ajudaram durante a minha formação, pelo carinho e apoio que tem me dado sempre.

A Teresa Chitula Luhuenba, por ser uma das pessoas que tem me dado força e motivação para poder chegar até onde almejo chegar, pelo amor e pelo carinho que tem me dado.

Aos meus irmãos pela constante inspiração, apoio incondicional e pelo vínculo familiar que enriquece minha jornada acadêmica. Sua presença e incentivo foram essenciais, tornando cada desafio mais leve e cada conquista mais significativa. Sou grato por poder contar com a amizade e o suporte de vocês ao longo deste percurso.

A Direcção do Instituto Superior de Angola (ISA), expresso minha profunda gratidão por proporcionar um ambiente educacional.

Ao meu orientador Edivaldo de Carvalho, pelo apoio que me deu durante a elaboração do presente trabalho.

A todos os meus colegas e amigos que de forma direta ou indireta me apoiaram para que eu pudesse chegar até a conclusão da minha licenciatura.

EPÍGRAFE

“Códigos que não têm bugs são códigos que nunca
foram escritos”

“David Parnas, 1970”

RESUMO

O presente trabalho tem como o tema “Implementação e Concepção de um Software para Gestão de Apartamentos para a empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA”. Com a seguinte problemática: Como a implementação de um software de gestão de apartamentos pode otimizar eficientemente os processos de administração de propriedades na empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA?. Tem como hipótese, implementação do software de gestão de apartamentos na XBXB Segundo Comércio e Prestação de Serviço (SU), LDA, resultará em uma melhoria significativa na eficiência dos processos de administração de propriedades. O objectivo geral visa em Desenvolver um software para a gestão de apartamentos para a empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA, com o intuito de otimizar e automatizar todos os processos envolvidos na gestão de imóveis, desde o cadastro até a administração, o específicos consistem em Identificar e analisar os principais processos envolvidos na gestão de apartamentos. Apresenta uma pesquisa Descritiva, com o modelo de abordagem Quali – Quantitativa. Trabalhamos com uma população de 15 trabalhadores, que equivale a 100% dos entrevistados. Foi possível perceber que a gestão de apartamentos é um mercado muito competitivo e cada vez mais surgem as necessidades de automatizar todos os processos relacionadas à gestão de apartamentos para proporcionar um ambiente de trabalho eficiente, gerando resultados e satisfação dos clientes. Dessa forma sugerimos que a capacitar os trabalhadores concernentes à utilização de meios tecnológica, a fim de facilitar a utilização do software para gestão de seus apontamentos.

Palavras-chave: Concepção, Implementação, Software, Gestão.

ABSTRACT

The theme of this work is “Implementation and Design of Software for Apartment Management for the company XBXB Segundo Comércio e Prestação de Serviços (SU), LDA”. With the following problem: How can the implementation of apartment management software can efficiently optimize property management processes in the company XBXB Segundo Comércio e Prestação de Serviços (SU), LDA?. It is hypothesized that the implementation of apartment management software at XBXB Segundo Comércio e Prestação de Serviços (SU), LDA, will result in a significant improvement in the efficiency of property management processes. The general objective aims to develop software for apartment management for the company XBXB Segundo Comércio e Prestação de Serviços (SU), LDA, with the aim of optimizing and automating all processes involved in property management, from registration to administration, the specifics consist of Identifying and analyzing the main processes involved in apartment management. It presents a descriptive research, with the Quali – Quantitative approach model. We work with a population of 15 workers, which is equivalent to 100% of those interviewed. It was possible to realize that apartment management is a very competitive market and there is an increasing need to automate all processes related to apartment management to provide an efficient work environment, generating results and customer satisfaction. Therefore, we suggest that workers be trained in the use of technological means, in order to facilitate the use of software to manage their notes.

Key words: Conception, Implementation, Software, Management.

LISTA DE ABREVIATURAS E SIGLAS

JAD- Joint Application Development

IFQ- Implantação da Função de Qualidade

UML- Unified Modeling Language

DER- Diagrama de Entidade – Relacionamento

ER- Entidade – Relacionamento

SGBD- Sistema de Gestão de Bases de Dados

SQL- Structured Query Language

DDL- Linguagem de Definição de Dados

DML- Linguagem interativa de manipulação de dados

ANSI- American National Standard Institute

C#- Csharp

ÍNDICE DE TABELAS

Tabela 1. Definição de Engenharia de Software.	7
Tabela 2. Requisitos Funcionais.	19
Tabela 3. Requisitos não funcionais.....	20
Tabela 4: Ciclo de vida do Banco de Dados.....	29
Tabela 5. Orçamento do Projeto	45

ÍNDICE DE FIGURAS

Figura 1 – Camadas da Engenharia de Software.....	7
Figura 2: Estágios do modelo de processo de software em cascata.....	11
Figura 3. Atividades da Engenharia de Requisitos.....	14
Figura 4. Processo de levantamento e análise de requisitos	16
Figura 5. Como Realmente funciona um projeto.	22
Figura 6. Classe	24
Figura 7. Diagrama de classe.....	24
Figura 8. Diagrama de objetos	25
Figura 9. Caso de Uso	26
Figura 10. Classe	27
Figura 11. Objecto.....	28
Figura 12. Diagrama e de Relacionamento Entidade	30
Figura 13. Tela de processamento.....	34
Figura 14. Tela de Login	34
Figura 15. Menu do Sistema.	35
Figura 16. Cadastramento de Usuário	36
Figura 17. Tela de Disponibilidade.	37
Figura 18. Tela de Cadastramento de Cliente.....	39
Figura 19. Tela de Contratos.....	40
Figura 20. Tela de Finança.....	41
Figura 21. Tela de Pagamento de Renda.	42

ÍNDICE

INTRODUÇÃO	1
CAPITULO I. FUNDAMENTAÇÃO TEÓRICA.....	5
1.1. Definições de termos e conceitos.....	5
1.2. Engenharia de Software	6
1.3. Processo de Software	8
1.3.1. Especificação de Software	9
1.3.2. Projeto e implementação de Software.....	9
1.3.3. Validação de Software	9
1.3.4. Evolução de Software	10
1.3.5. Modelo de Processo de Software	10
1.4. Engenharia de Requisitos	12
1.4.1. Estudo de viabilidade	13
1.4.2. Levantamento e análise de requisitos	15
1.4.3. Documentação dos requisitos	16
1.4.4. Validação dos requisitos	17
1.4.5. Requisitos Funcionais e não Funcionais	18
1.4.5.1. Requisitos Funcionais.....	18
1.4.5.2. Requisitos Não Funcionais	19
1.5. Unified Modeling Language (UML)	21
1.5.1. Diagrama Caso de uso.....	22
1.5.2. Diagrama de Classes	22
1.5.3. Diagrama de objetos	24
1.5.4. Diagramas do sistema.....	25
1.5.4.1. Caso de Uso	25
1.5.4.2. Classe	25

1.5.4.3. Objecto.....	26
1.6. Banco de Dados	27
1.6.1. Diagrama de entidade – Relacionamento (DER)	27
1.6.2. Sistemas de Gestão de Bases de Dados (SGBD)	28
1.7. Projeto de Banco de Dados.....	29
1.8. SQL	30
1.9. Linguagem de Programação C#.....	31
1.10. Telas e Designer do Software	31
1.10.1. Tela de Processamento	31
1.10.2. Tela de login	32
1.10.3. Tela de menu do sistema.....	33
1.10.4. Tela de cadastro de Usuários	34
1.10.5. Tela de Disponibilidade	35
1.10.6. Tela de Cadastramento de Clientes.....	36
1.10.7. Tela de Gestão de Contrato.....	37
1.10.8. Tela de Finanças	38
1.10.9. Tela de Pagamento de Renda	39
CAPÍTULO II. METODOLOGIA DE PESQUISA.....	40
2.1. Tipos de Pesquisa.....	40
2.2. Modelos de Abordagem	40
2.3. Nível de estruturação de método	40
2.4. Técnicas de recolha de dados.....	41
2.5. Instrumentos e procedimentos de recolha de dados.....	42
2.6. População e amostra.....	42
2.7. Dificuldades encontradas	42
CAPÍTULO III. DISCUSSÃO E RESULTADOS.....	43

3.1. Resultados da Implementação	43
3.2. Recursos Necessários para o Desenvolvimento do Projeto.....	43
3.3. Estimativa de Custos.....	44
3.4. Orçamento do Projeto	44
3.5. Discussão dos resultados.....	45
CONCLUSÃO.....	46
SUGESTÕES	47
REFERÊNCIAS BIBLIOGRÁFICAS.....	48

APÊNDICES

ANEXOS

1. INTRODUÇÃO

Nos últimos anos, a gestão de imóveis tem se tornado cada vez mais complexa, soluções tecnológicas avançadas para aperfeiçoar os processos e melhorar a eficiência das empresas do setor. Nesse contexto, a XBBB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, empresa focada na prestação de serviços e esta ligado ao mercado imobiliário com um projecto de imoveis para estudantes que estejam a frequentar o ensino superior em Luanda, e que tem buscado constantemente inovações para se destacar em um mercado altamente competitivo.

Este projecto focaliza na implementação e concepção de um software de gestão de apartamentos adaptado às necessidades específicas da empresa XBBB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, visando a fornecer aluguer de apartamentos a estudantes que estejam a frequentar o ensino superior.

Uma das áreas que apresenta maior desafio na gestão de imóveis é a gestão de apartamentos, pois envolve diversos processos que exigem organização e eficiência. Desde a captação de imóveis, passando pela negociação e fechamento de contratos, a manutenção e administração dos imóveis, até tudo precisa ser gerada a satisfação dos clientes e aumento dos lucros.

Surge a necessidade de um software que possa automatizar os processos de gestão de apartamentos para a XBBB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, tornando-os mais ágeis, precisos e eficientes. O presente trabalho tem como objectivo de desenvolver um software para a gestão de apartamentos da empresa, que permita o gerenciamento completo do ciclo de vida dos imóveis, desde a sua entrada na carteira da empresa até a sua saída.

O software deverá contemplar funcionalidades como cadastro de imóveis, negociação e fechamento de contratos, gestão de manutenção e reparos, acompanhamento de vistorias e inspeções, entre outras.

A empresa XBBB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, situada na Rua Rosa das Neves, Bairro Kikuxi, município de Viana, província de Luanda, com NIF: 500861804 enfrentam desafios diários em relação à gestão de seus imóveis, desde o cadastro até a manutenção e administração, o que pode levar

a atrasos na entrega de imóveis aos clientes, perda de informações importantes e falta de eficiência operacional, que leva-nos a levantar a seguinte pergunta de partida:

“Como a implementação de um software de gestão de apartamentos pode otimizar eficientemente os processos de administração de propriedades na empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA?”.

Hipóteses

“Segundo Baptista e Sousa (2011, p. 26), “as hipóteses são uma resposta prévia ao problema proposto e, habitualmente, são desenvolvidos com base em estudos anteriormente realizados de acordo com tema escolhido”, pelo que a “hipótese deve justificar o trabalho da parte empírica da investigação” (Hill e Hill, 2002, p. 22).

- ✓ H1: A eficiência operacional da administração de propriedades na empresa XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, apresenta diferença estatisticamente significativa entre a implementação do software de gestão de apartamentos e os métodos tradicionais baseados em documentos.
- ✓ H2: A implementação do software de gestão de apartamentos na XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, resultará em uma melhoria significativa na eficiência dos processos de administração de propriedades em comparação com os métodos tradicionais baseados em documentos.

Objetivos de Estudo

Objetivo de estudo é um enunciado declarativo que precisa a orientação da investigação segundo o nível dos conhecimentos estabelecidos no domínio em questão. “Especifica as variáveis-chave, a população alvo e o contexto de estudo” (Fortin, 2009, p. 100).

Geral

- ✓ Desenvolver um software para a gestão de apartamentos para a empresa XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, com o intuito de otimizar e automatizar todos os processos envolvidos na gestão de imóveis, desde o cadastro até a administração.

Específicos

- ✓ Identificar e analisar os principais processos envolvidos na gestão de apartamentos.
- ✓ Levantar e definir os requisitos funcionais e não funcionais do software, considerando as necessidades específicas da empresa.
- ✓ Definir a arquitetura do software, levando em consideração as melhores práticas de desenvolvimento de software e as necessidades da empresa.

Delimitação e limitação

- a) Delimitamos a nossa pesquisa na empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA, situada em Luanda, no Município de Viana, Rua Rosa das Neves, 2024.
- b) Limitamos a pesquisa na área de gestão de apartamentos da empresa XBXB Segundo Comércio e Prestação de Serviço (SU), LDA.

Justificativa

A justificativa compreende a apresentação de forma clara e objetiva das razões de ordem teórica e ou prática que fundamentam a pesquisa. Justificam-se a escolha do tema, a delimitação realizada e a relação que o pesquisador possui com ele. “Procura-se aqui demonstrar a legitimidade, a pertinência, o interesse e a capacidade do aluno em lidar com o referido tema” (CERVO & BERVIAN, 2002, p. 127).

A criação de um software para gestão de imóveis é uma tendência no mercado imobiliário atual, e empresas que adotam essas soluções têm uma vantagem competitiva em relação às que ainda utilizam processos manuais. Com o software, a XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, pode se destacar no mercado imobiliário e aumentar sua participação no setor.

Nesse contexto, a concepção e implementação de um software para gestão de apartamentos para a XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA, é fundamental para otimizar e automatizar todos os processos envolvidos na gestão de imóveis, partindo do cadastro até a manutenção e administração. Com a implementação de um software, a empresa pode aumentar sua eficiência

operacional, reduzir seus custos e aumentar seus lucros, garantindo a satisfação dos clientes.

Estrutura do Trabalho

O presente trabalho de pesquisa com o tema, Implementação e Concepção de um Software para Gestão de Apartamentos da Empresa XBXB Segundo Comércio Geral e Prestação de Serviço (SU), LDA. Está estruturado da seguinte forma:

Introdução do tema em destaque, problemática do tema, hipóteses que trazem as possíveis respostas da pergunta levantada, os objectivos de estudo, delimitação e Justificativa do tema.

- **Capítulo I.** Fundamentação Teórica, onde temos as definições de termos e conceitos, e outros temas.
- **Capítulo II.** Metodologia de Pesquisa foi apresentada os materias e métodos usados na pesquisa.
- **Capítulo III.** Apresentação e Discunsão de Resultados.

No final temos as Conclusões, Sugestões e as Referências Bibliográficas.

CAPITULO I. FUNDAMENTAÇÃO TEÓRICA

1.1. Definições de termos e conceitos

a) Implementação

"A implementação pode ser definida como o processo de colocar em prática uma estratégia, política ou programa, que envolve a execução de atividades específicas, a alocação de recursos necessários e o acompanhamento do progresso para alcançar os resultados desejados" (BIRKINSHAW & GODDARD, 2009, pág. 305).

"A implementação é a fase crítica de transformar uma ideia em ação tangível, abrangendo a tradução de conceitos em práticas concretas, a capacidade de recursos, a gestão de mudanças e a superação de desafios para alcançar os objetivos pretendidos" (BRESSER-PEREIRA, 2004, pág. 28).

"A implementação refere-se ao processo de transformar planos políticos em ação real, envolvendo a execução de tarefas, a coordenação de recursos humanos e materiais, e adaptação contínua para garantir a eficácia da implementação" (Ostrom, 1976, p. 510).

b) Software

Segundo Sommerville (2007), software é o programa, mas também a documentação e configuração associadas e necessárias para que o programa opere corretamente.

De acordo com Stair e Reynolds (2011), o software consiste em programas que comandam a operação do computador. Esses programas permitem que o computador processe as folhas de pagamento, envie as contas para os clientes e forneça aos gerentes informações para aumentar os lucros, reduzir custos e oferecer melhor serviço ao consumidor. Com o software, as pessoas podem trabalhar a qualquer hora em qualquer lugar.

Segundo Paula Filho (2001), o software é a parte programável de um sistema de informática. Ele é um elemento central: realiza estruturas complexas e flexíveis que trazem funções, utilidade e valor ao sistema. Mas outros componentes são indispensáveis: as plataformas de hardware, os recursos de comunicação de

informação, os documentos de diversas naturezas, as bases de dados e até manuais que se integram aos automatizados.

c) Gestão

Segundo Pires (2007), a palavra gestão é uma palavra polissêmica. Em conformidade, é necessário adaptá-la ao ambiente onde está a ser utilizada de maneira a assumir em termos relativos o seu verdadeiro significado.

A gestão pode ser entendida como o processo de coordenação e integração de recursos, tendente à consecução dos objetivos estabelecidos, através do desempenho das atividades de planejamento, organização, direção e controle. Pode ser também assimilada a processo de trabalho com e através dos outros, a fim de se atingirem eficazmente os objetivos organizacionais traçados, utilizando-se eficientemente os recursos escassos, num contexto em constante mutação (Santos, 2008).

Para Chiavenato (2004), a Gestão é a “maneira de governar Organizações ou parte delas”. É o processo de planejar, organizar, dirigir e controlar o uso de recursos organizacionais para alcançar determinados objetivos de maneira eficiente e eficaz

1.2. Engenharia de Software

O conceito de Software é percebido como um conjunto de programas e procedimentos, que processam dados gerando informações ao hardware. A importância e complexidade dos Softwares foram aumentando ao longo dos anos,

“As primeiras aplicações de Software dos computadores em empresas foram desenvolvidas no início da década de 1950. O Software era menos importante (e menos caro) nos sistemas de computação da época. Hoje, o Software representa uma porcentagem muito maior do custo dos sistemas de computação modernos. O preço do Software aumentou assustadoramente.”
(TURBAN, JR. E POTTER, 2007, P. 318)

A partir desta modernização e aumento dos custos de um Software veio a necessidade de uma nova maneira de se desenvolver um Software.

Segundo Sommerville (2007, p. 3) “o conceito de Engenharia de Software foi inicialmente proposto em 1968, em uma conferência organizada para discutir o que foi então chamado de “crise de software”“. A engenharia surgiu como uma

ferramenta auxiliar na construção de um Software, valorizando e tornando-os mais eficientes, como cita Tonsig (2008, p. 65) “objetos que são frutos do trabalho de engenharia, constituem-se soluções economicamente viáveis para problemas práticos, obtidas do conhecimento científico sobre um determinado domínio tecnológico, que podem partilhar soluções anteriores.” A Engenharia de Software pode ser definida da seguinte maneira:

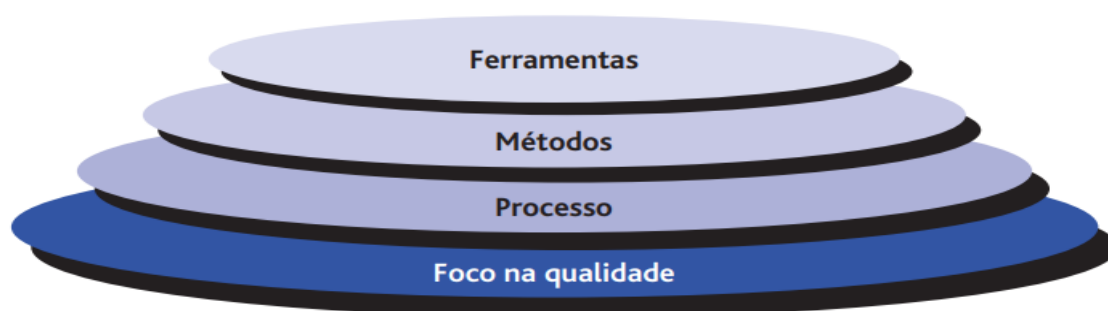
Tabela 1. Definição de Engenharia de Software.

Sommerville (2007, p. 5)	“Engenharia de Software é uma disciplina de engenharia relacionada a todos os aspectos de produção de software, desde os estágios iniciais de especificação do sistema até sua manutenção, depois que este entrar em operação.”
Pressman (2011)	Engenharia de Software é a utilização dos recursos e princípios da engenharia em função de obter um Software confiável e que funcione eficientemente em uma máquina real.
Pfleeger (2004)	A Engenharia de Software é responsável por agrupar os conhecimentos teóricos e funções computacionais da Ciência da Computação, juntamente com o problema proposto pelo cliente a fim de se obter ferramentas e técnicas para resolver o problema.

Fonte: Tonsig (2008, p. 65).

Segundo Pressman (2016, p.15), a Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade, conforme demonstrado na Figura 1.

Figura 1 – Camadas da Engenharia de Software



Fonte: Pressman (2016, p.16).

Pressman (2016) define a qualidade, como item preponderante para o sucesso de um software, e vários estudos determinam que ela esteja intrinsicamente ligada ao atendimento à requisitos. A camada de processo constitui o pilar para o controle e gerenciamento do trabalho de desenvolvimento. A camada de métodos oferece subsídios técnicos para as tarefas desenvolvidas durante o ciclo de desenvolvimento, e por último na camada mais alta temos as ferramentas fornecendo automação das atividades de forma organizada e repetível.

1.3. Processo de Software

De acordo Pressman (2005), processo de software é uma metodologia para atividades, tarefas e ações necessárias para o desenvolvimento de um software com qualidade. Conforme Sommerville (2010), processo de software é um grupo de atividades que conduzem o processo de produção e desenvolvimento de um produto de software. Essa produção pode ser de um novo software, ou de produtos que já existem, mas precisam passar por modificações ou evoluções. Existe mais de um tipo de processo de software, mas todos os processos incluem quatro atividades fundamentais (SOMMERVILLE, 2010):

- Especificação de software, no qual as funcionalidades e restrições do funcionamento do produto são definidas;
- Projeto e implementação de software, produção do produto atendendo as especificações;
- Validação de software, validação do projeto para garantir que o produto atende às demandas do cliente;
- Evolução de software, evolução do produto para que atenda às necessidades de mudança do cliente.

Processos de software são complexos, dependem de pessoas para tomar decisões. Eles evoluem na tentativa de obter a melhor capacidade das pessoas e organizações envolvidas no desenvolvimento de um sistema. Para sistemas críticos, por exemplo, em que uma falha pode levar a perdas significativas de dinheiro, ameaças ao ser humano ou danos físicos, o processo precisa ser muito bem estruturado (DROBKA; NOFTZ; RAGHU, 2004).

1.3.1. Especificação de Software

A atividade de especificação de software, ou engenharia de requisitos, representa a etapa de levantamento de requisitos do sistema, em que se compreende e identifica restrições relacionadas operação e desenvolvimento do sistema. Esta etapa é crítica, pois erros causados no levantamento de requisitos podem ser propagados para as outras atividades do processo de software, gerando problemas na implementação e validação do sistema (SOMMERVILLE, 2010).

Esta atividade tem como objetivo a construção da documentação de requisitos do sistema. Estes requisitos são especificações acordadas com os responsáveis pelo projeto, especificações que satisfazem as necessidades do cliente.

1.3.2. Projeto e implementação de Software

A atividade de projeto e implementação de software é o estágio de desenvolvimento do sistema descrito. É o estágio em que a descrição dos requisitos definidos na atividade de especificação é convertida na implementação do sistema de fato, em um produto funcional (SOMMERVILLE, 2010).

O projeto de software é a documentação da estrutura do software, ou seja, descrição dos modelos, estruturas, interfaces e componentes necessários para a implementação do software. Essa documentação pode ser feita através de diagramas de classes, e ocorre durante todo o desenvolvimento do sistema, que é feito de forma iterativa, passando por revisões constantes e correções. O desenvolvimento do projeto durante essa atividade é a programação do sistema, onde todos os itens descritos na documentação do projeto de software se tornam o produto funcional.

1.3.3. Validação de Software

A atividade de validação de software, também conhecida como verificação e validação, tem como objetivo a execução de testes em cima do sistema desenvolvido. Esses testes servem para validar que o sistema satisfaz todas as especificações levantadas pelo cliente na atividade de especificação do software (SOMMERVILLE, 2010).

A validação e verificação do sistema desenvolvido não ocorrem apenas no final do projeto, após o desenvolvimento de todo o sistema, ocorre durante todo o processo de desenvolvimento. Como o desenvolvimento é um processo iterativo, a validação também, testando toda entrega de pequenas porções funcionais do sistema, assim como a validação da integração dessas novas entregas com o que já foi desenvolvido em iterações anteriores.

1.3.4. Evolução de Software

A atividade de evolução do software representa a evolução e manutenção de um sistema. Mudanças em software são consideradas alterações muito mais baratas do que alterações em hardware, por exemplo. Essas podem ser feitas a qualquer momento, durante ou após a finalização do desenvolvimento (SOMMERVILLE, 2010).

A evolução e manutenção de software devem ser vista como um processo contínuo, junto com o desenvolvimento do projeto. Por se tratar de um processo evolutivo, as especificações e requisitos do cliente podem ser alterados, prolongando o período de vida do produto, que segue se adaptando às necessidades do cliente.

Evoluir o software envolve também a resolução de débitos técnicos do projeto. Débito técnico ou dívida técnica, de acordo com Brown (2010), representa a dívida que um projeto de software assume quando toma a decisão de priorizar a entrega, ao invés da qualidade. Isso acontece quando requisitos complexos do projeto de software são solucionados de maneira inadequada, exigindo que a solução seja revisitada no futuro para que o projeto possa evoluir da maneira correta.

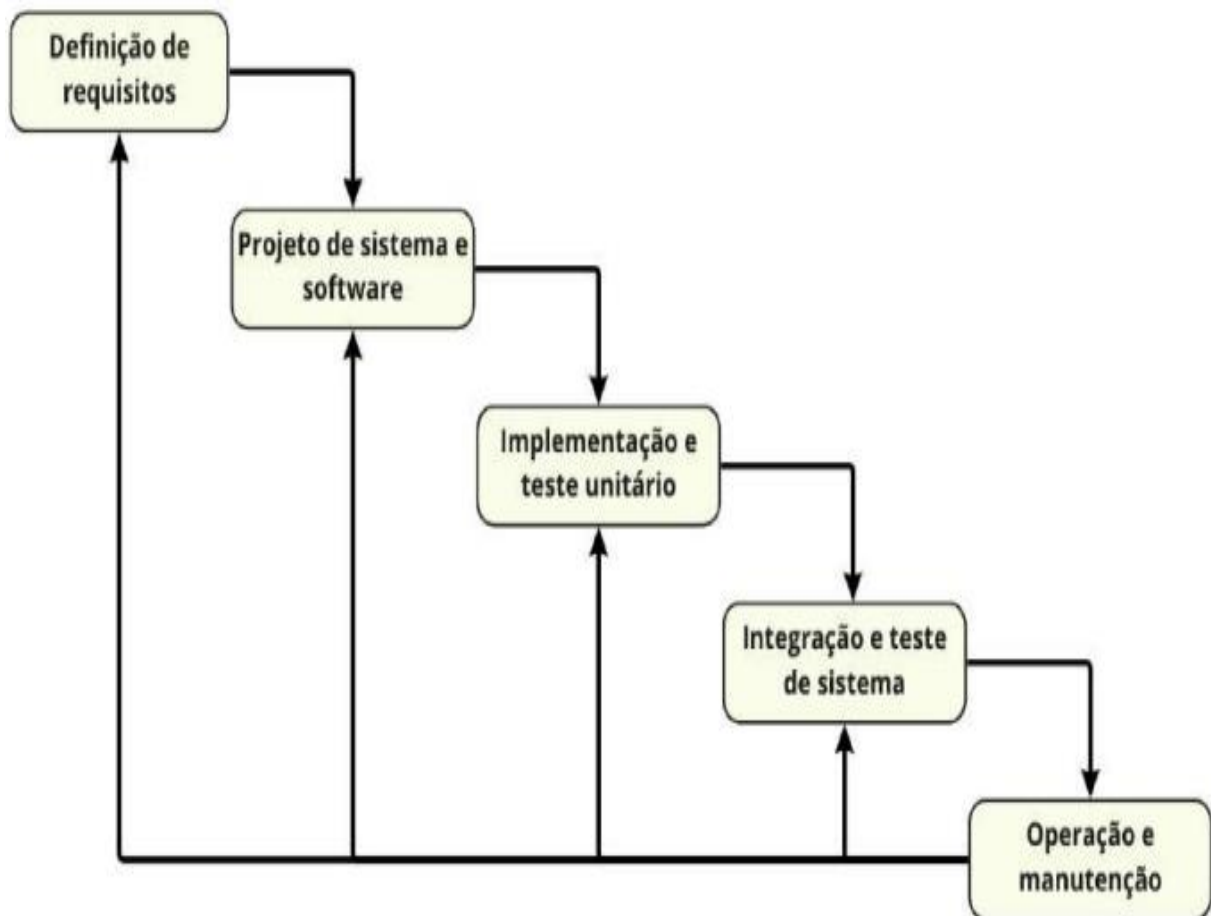
1.3.5. Modelo de Processo de Software

Os processos de software, de acordo com Sommerville (2010), são representados como modelos, e cada modelo apresenta suas próprias características. Esses modelos são estruturas que podem dividir as atividades, que representam uma visão macro do processo de software. Pensa-se que cada atividade complexa pode ser dividida em tarefas de menor complexidade, expandindo o processo de software.

Um exemplo de modelo de processo de software é o modelo em cascata. No modelo em cascata, as atividades são separadas em estágios que seguem uma ordem restrita, formando assim a cascata, como mostra a Figura 2. No modelo cascata, os principais estágios do desenvolvimento são representados da seguinte forma:

- *Análise e definição de requisitos*, estágio de estabelecimento das restrições e metas do sistema a ser desenvolvido. Nesse estágio são definidas as especificações do sistema a partir do levantamento de requisitos.

Figura 2: Estágios do modelo de processo de software em cascata.



Fonte: Reproduzido de Sommerville (2010).

- *Projeto de sistema e software*, estágio em que os requisitos são analisados para alocar as necessidades do sistema em relação à tecnologias que serão utilizadas no desenvolvimento. Essas necessidades também envolvem a definição da arquitetura do sistema.

- *Implementação e teste unitário*, estágio em que o projeto é desenvolvido e, em conjunto com o desenvolvimento, são feitos testes unitários. O teste unitário nesse estágio serve para a verificação de que cada unidade entregue corretamente as necessidades levantadas no estágio de definição do sistema.
- *Integração e teste de sistema*, estágio em que as unidades do sistema desenvolvidas, de forma iterativa na etapa de implementação, são integradas com o sistema por completo. Os testes de sistema nessa etapa servem para verificar se todas as unidades, quando utilizadas em conjunto como um sistema completo, satisfazem os requisitos do estágio de definição, e após essa validação, o sistema pode ser entregue ao cliente.
- *Operação e manutenção do sistema*, etapa em que o sistema entra em uso pelo cliente final. Nessa etapa, a manutenção do sistema ocorre conforme erros, que não foram encontrados durante os testes, são encontrados. Esses erros são corrigidos e melhorias são levantadas conforme as novas necessidades do cliente, descobertas a partir do uso do sistema.

1.4. Engenharia de Requisitos

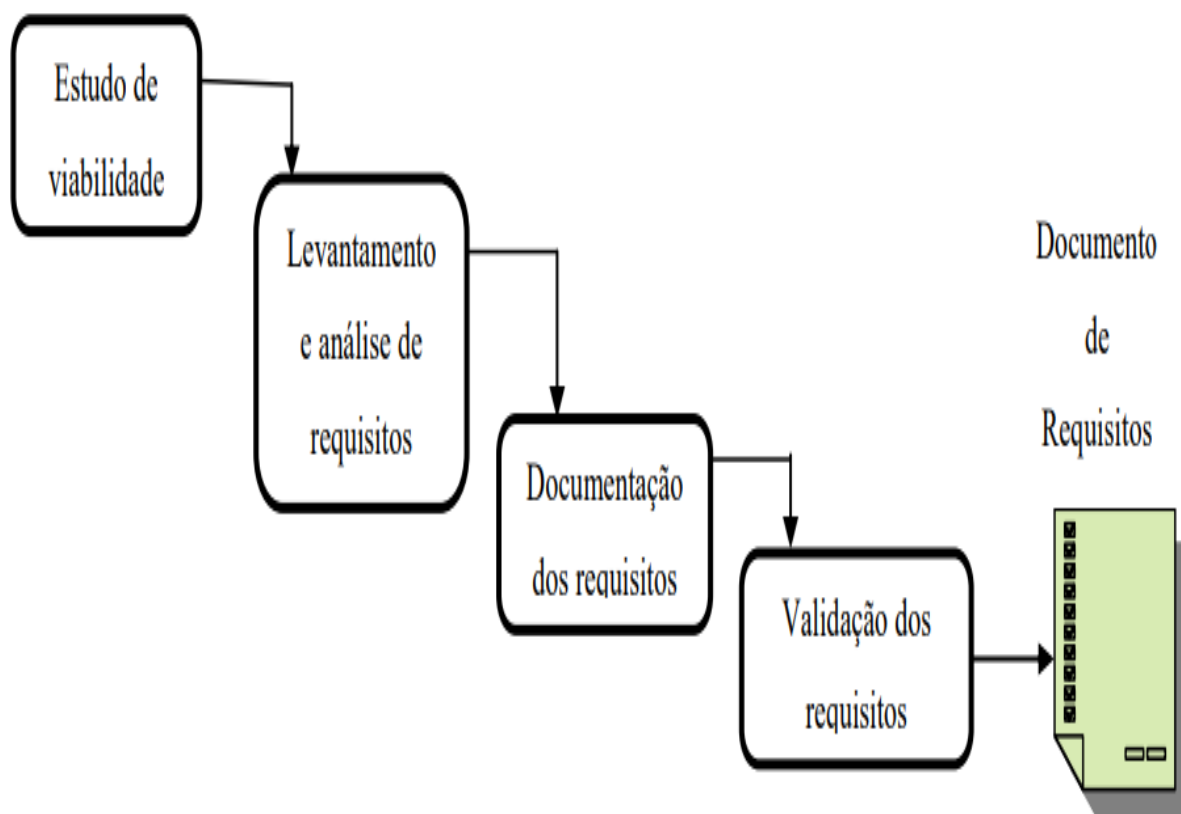
A engenharia de requisitos, de acordo com Sommerville (2007, p. 49), tem como objetivo definir o que o sistema deve fazer, quais as necessidades reais e identificar quais restrições existem para que o software seja desenvolvido. É nesse processo da engenharia de software que ocorre a comunicação entre o cliente e o analista da equipe de desenvolvimento. Quando essa comunicação não é bem sucedida, o restante do projeto pode ficar comprometido, causando impacto no custo e prazo.

No entanto, existem diversas técnicas que são responsáveis em se obter requisitos de maneira mais adequada. Propondo demonstrar qual técnica é mais apropriada para determinada situação, este trabalho descreve as técnicas existentes e realiza uma análise comparativa das principais técnicas.

Para definir quem são os participantes da engenharia de requisitos, Paula Filho (2003, p. 4), caracteriza como cliente aquele que tem como função financiar o projeto, conhecido também como patrocinador. Já aquele que usa efetivamente o sistema é denominado de usuário. No entanto, podem ocorrer situações em que o próprio usuário é o cliente.

Conforme pode ser observado na Figura 2, Sommerville (2007, p. 50) define que o processo de engenharia de requisitos é composto de quatro atividades: estudo de viabilidade, levantamento e análise de requisitos, documentação dos requisitos e, por fim, validação dos requisitos. Ao final dessas atividades, é obtido o documento de requisitos.

Figura 3. Atividades da Engenharia de Requisitos



Fonte: (SOMMERVILLE, 2007, p. 50).

1.4.1. Estudo de viabilidade

A atividade de estudo de viabilidade é o primeiro passo para o desenvolvimento do software. É nesse momento que ocorre uma primeira comunicação entre o cliente e o analista da equipe de desenvolvimento. Conforme Sommerville (2007, p. 97), os resultados dessa atividade definem se o sistema a ser construído contribui ou não para a organização.

Segundo Pressman (2006, p. 118), o analista, juntamente com o cliente, tentam identificar qual o tamanho e a funcionalidade do sistema, realizam um estudo de viabilidade, por fim obtendo um documento denominado de escopo do projeto. Tal documento possivelmente servirá de base para o restante do projeto. É

importante ressaltar que todas essas informações estão sujeitas a mudanças, pois a percepção do cliente pode sofrer alterações bem como novas tecnologias podem surgir.

Sommerville (2007, p. 97) afirma que para se obter as informações do escopo do projeto, é necessário que possíveis perguntas sejam respondidas pelo cliente, por exemplo:

- a) Como a organização se comportaria se esse sistema não fosse implementado?
- b) Quais são os problemas com os processos atuais e como o novo sistema ajudaria a reduzir esses problemas?
- c) Qual será a contribuição direta do sistema para os objetivos e requisitos da empresa?
- d) As informações podem ser transferidas e recebidas de outros sistemas da organização?
- e) O sistema requer tecnologia que ainda não foi usada na organização?

Após respondidas a primeira e segunda questões, o analista terá a capacidade de determinar qual a importância do sistema para a organização, expondo seu funcionamento para o cliente, de modo que no final o produto solicitado otimize os processos da empresa. Por meio da terceira questão será possível ter conhecimento de como o sistema pode contribuir para os objetivos da organização. A informação obtida na quarta questão verifica se o software irá realizar comunicação com outros sistemas existentes. Neste caso, uma avaliação de complexidade poderá ser realizada. A quinta questão determina em qual tecnologia o sistema será desenvolvido e então verificar o seu uso no ambiente do cliente

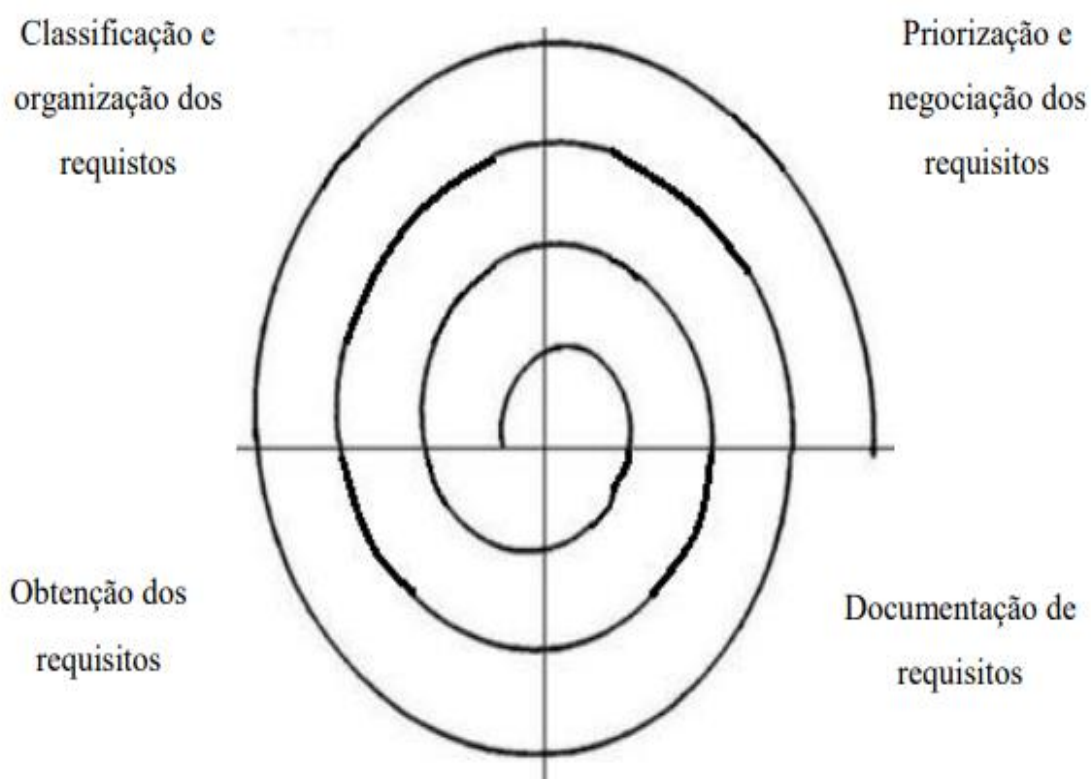
Por fim, depois de identificadas as informações, é criado o escopo do projeto, que deve ser revisado por todos os participantes. Conforme Sommerville (2007, p. 97), tal documento deve conter uma recomendação se o sistema deve ou não prosseguir para seu desenvolvimento. Caso ele contribua para os objetivos da organização, é iniciada a fase de levantamento e análise de requisitos, a qual é descrita a seguir.

1.4.2. Levantamento e análise de requisitos

Essa atividade realiza o levantamento das necessidades do usuário, isto é, os requisitos do sistema, por meio de diversas técnicas, conforme pode ser observado. Para Sommerville (2007, p. 98), a atividade de análise de requisitos visa priorizar e resolver conflitos entre requisitos, pois quando vários usuários participam desse processo, é inevitável que ocorra contradição entre requisitos levantados de usuários distintos.

Conforme proposto por Sommerville (2007, p. 98), o modelo de processo comum para a fase de levantamento e análise de requisitos é demonstrado na Figura 3.

Figura 4. Processo de levantamento e análise de requisitos



Fonte: (SOMMERVILLE, 2007, p. 98).

Na atividade de levantamento de requisitos, o analista inicia uma comunicação com o usuário utilizando técnicas para que possa obter o conhecimento das necessidades do usuário. Com as respostas obtidas, é possível identificar quais serviços o sistema deve oferecer, quais as suas restrições, o que é

esperado pelo usuário e demais informações, tal como a possibilidade de integração com outros sistemas.

Existem diversas técnicas para realizar o levantamento dos requisitos, tais como *Joint Application Development* (JAD), prototipação, entrevista, questionário, observação, Implantação da Função de Qualidade (IFQ), casos de uso e pontos de vista. Como um exemplo, na técnica de entrevista, segundo Sommerville (2007, p. 101), o analista formula questões para os usuários sobre o sistema que eles usam e o sistema a ser desenvolvido. As entrevistas são úteis para se obter um entendimento geral sobre o que os usuários fazem, como eles irão interagir com o sistema e quais as dificuldades que enfrentam com os sistemas atuais.

Como outro exemplo, na prototipação, conforme Pressman (2006, p. 42), o analista e o usuário definem os objetivos gerais e um protótipo é criado e apresentado ao cliente. Esse processo pode ocorrer por diversas vezes, até que seja satisfeita a necessidade do usuário.

1.4.3. Documentação dos requisitos

Depois de realizado a etapa de levantamento e análise de requisitos, a tarefa seguinte é criação do documento de requisitos. Segundo Pressman (2006, p. 120), uma especificação pode ser um documento escrito, um modelo gráfico, ou matemático formal, um protótipo ou qualquer combinação desses documentos.

Procurando estabelecer uma consistência na especificação a ser criada, Pressman (2006, p. 120) sugere uma criação de um documento de requisitos padrão. Entretanto, essa padronização é mais bem adequada quando aplicada em sistemas grandes, uma vez que em sistemas de pequeno ou médio porte é necessário ocasionalmente ser flexível.

A especificação de requisitos é escrita pelo analista, contendo as informações que foram coletadas no levantamento dos requisitos. Segundo Pressman (2006, p. 120), a especificação é o documento de trabalho que servirá como referência para as demais atividades da engenharia de software. O documento contém informações sobre o que o sistema deve fazer, quais as necessidades reais e quais restrições existem para que o software seja desenvolvido.

1.4.4. Validação dos requisitos

A validação dos requisitos, segundo Sommerville (2007, p. 105) tem como objetivo garantir que a necessidade real do usuário esteja descrita corretamente no documento de especificação dos requisitos. A validação é extremamente importante, pois o custo para correção de um requisito nessa fase é bem inferior ao custo nas fases posteriores, como implementação ou testes. Isso ocorre pois se defeitos encontrados nessas fases, os requisitos devem ser novamente levantados e posteriormente são implementados e testados.

Pressman (2006, p. 120) afirma que os requisitos devem ser examinados para que sejam encontradas inconsistências, ambiguidades e omissões. Para tal atividade, Sommerville (2007, p. 106) propõe algumas técnicas de validação, tais como revisões de requisitos, prototipação e geração de casos de teste.

Como um exemplo, a revisão de requisitos pode ser formal ou informal. A revisão formal é um processo no qual o analista detecta e apresenta para o cliente os possíveis problemas encontrados, como inconsistências, omissões e contradições. Posteriormente, as dúvidas levantadas são discutidas entre a equipe de desenvolvimento e o cliente, tendo como objetivo uma solução para o problema.

A revisão de requisitos informal é um debate que ocorre entre analista e cliente, no qual muitos problemas podem ser identificados sem que exista a necessidade de avançar para a revisão formal. Como um outro exemplo, na prototipação, os requisitos são levantados e são então apresentadas telas para o usuário, tornando fácil a compreensão do sistema a ser desenvolvido. A motivação por trás da prototipação é a dificuldade do usuário em compreender como será o sistema baseado apenas na documentação dos requisitos.

Por fim, Sommerville (2007, p. 106) descreve outra técnica de validação de requisitos, denominada de geração de casos de testes. Essa técnica verifica se o requisito é passível de implementação. Caso contrário, o requisito deve ser reconsiderado, pois ele é extremamente difícil de ser interpretado e implementado. Segundo Sommerville (2007, p. 106), testes realizados antes da codificação do sistema é uma prática comum do Extreme Programming (XP), que é uma metodologia ágil de desenvolvimento de sistemas. Segundo Sommerville (2007, p. 265), os requisitos são expressos em cenários, também conhecidos como histórias

do usuário. Partindo das informações adquiridas, a equipe de desenvolvimento estima o esforço e recursos necessários para a implementação. O cliente atribui uma prioridade em cada cenário levantado tendo como referência a importância da funcionalidade para o sistema. Em seguida, Pressman (2006, p. 65) recomenda a criação de testes de unidade antes do início implementação, de forma que o desenvolvedor esteja melhor preparado para a codificação do sistema. Ademais, os programadores trabalham em pares, visando uma maior qualidade no desenvolvimento do produto.

Entretanto, conforme Sommerville (2007, p. 106), essa atividade não é simples. Com toda experiência dos analistas no desenvolvimento de sistemas, eles ainda encontram dificuldades em levantar os requisitos de forma a atender a necessidade do usuário. Quando realizada pelos próprios usuários, a propensão a erros nos requisitos é ainda maior. Assim, é inevitável que ocorra mudanças de requisitos nas fases seguintes do projeto.

Por fim, após a validação dos requisitos, é criado o documento de requisitos, o qual contém todas as informações necessárias para o desenvolvimento do sistema.

1.4.5. Requisitos Funcionais e não Funcionais

Abaixo serem ilustrados os requisitos funcionais e não funcionais do nosso caso prático (Software de Gestão de Apartamentos).

1.4.5.1. Requisitos Funcionais

Requisitos Funcionais são declarações de serviços que o sistema deve prover, descrevendo o que o sistema deve fazer (SOMMERVILLE, 2007). Um requisito funcional descreve uma interação entre o sistema e o seu ambiente (PFLEEGER, 2004), podendo descrever, ainda, como o sistema deve reagir a entradas específicas, como o sistema deve se comportar em situações específicas e o que o sistema não deve fazer (SOMMERVILLE, 2007).

Para o nosso trabalho de pesquisa, levantamos os seguintes requisitos funcionais de acordo a tabela 2. Requisitos Funcionais.

Tabela 2. Requisitos Funcionais.

Código	Agente	Descrição
R001	Todos Actualizadores	O sistema deve permitir fazer login.
R002	Administrador	O sistema deve permitir cadastrar, excluir e editar Usuários.
R003		O sistema deve permitir visualizar Clientes, Blocos, Apartamentos e Finanças.
R004		O sistema deve permitir cadastrar e excluir blocos Blocos.
R005		O sistema deve permitir visualizar contratos.
R006	Normal	O sistema deve permitir Cadastrar, visualizar e excluir Clientes.
R007		O sistema deve permitir cadastrar, visualizar e editar Quarto.
R008		O sistema deve permitir visualizar Bloco.
R009		O sistema deve permitir renovar e terminar contratos.
R010		O sistema deve permitir visualizar contratos.
R011		O sistema deve permitir efectuar pagamentos.
R012		O sistema deve permitir visualizar pagamentos.

Fonte: Elaboração próprio.

1.4.5.2. Requisitos Não Funcionais

Requisitos Não Funcionais descrevem as restrições sobre os serviços ou funções oferecidos pelo sistema (SOMMERVILLE, 2007), as quais limitam as opções para criar uma solução para o problema (PFLEEGER, 2004). Neste sentido, os

requisitos não funcionais são muito importantes para a fase de projeto (design), servindo como base para a tomada de decisões nessa fase.

Para o nosso projecto de pesquisa levantamos os seguintes requisitos não funcionais de acordo a Tabela 3.

Tabela 3. Requisitos não funcionais

Código	Tipo	Especificação de Requisitos
R013	Desempenho	- O sistema deve suportar um número específico de usuários simultâneos, garantindo resposta rápida mesmo em momentos de carga intensa.
R014	Segurança	- Implementar autenticação segura para administrador e secretário.
		- Garantir proteção dos dados sensíveis, utilizando criptografia.
R015	Usabilidade	- Interface intuitiva e amigável para facilitar o uso pelos administradores e secretários.
R016	Manutenção	- Facilidade de atualização e manutenção do software.
		- Registro de logs para rastreamento de atividades e resolução de problemas.
R017	Compatibilidade	- Garantir compatibilidade com diferentes ambientes do Windows.
R018	Desenvolvimento	- Adotar práticas de programação sustentável, promovendo a escalabilidade e a adaptabilidade do sistema.

Fonte: Elaboração próprio.

1.5. Unified Modeling Language (UML)

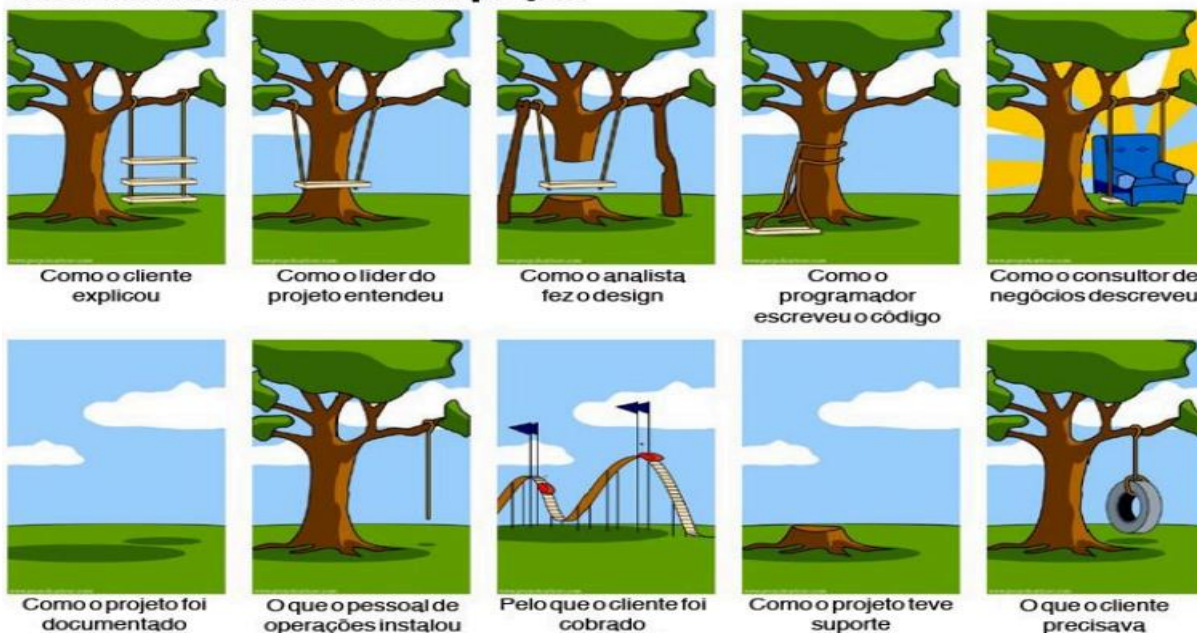
Primeiramente é importante salientar a importância da modelagem. Segundo os autores Booch, Rumbaugh e Jacobson (2005, p. 3), “uma empresa de Software bem sucedida é aquela que fornece Software de qualidade e capaz de atender às necessidades dos respectivos usuários.” Estes autores defendem que a modelagem tem papel fundamental nas atividades que levam a implantação de um bom Software.

Ainda segundo Booch, Rumbaugh e Jacobson (2005, p. 6) “a modelagem é uma técnica de engenharia aprovada e bem aceita. Construímos modelos de arquitetura de casas e de grandes prédios para auxiliar seus usuários a visualizar qual será o produto final.” Podendo-se dizer que a modelagem é uma simplificação da realidade e que a construção de modelos se dá no intuito de compreender melhor o sistema a ser desenvolvido.

Segundo Guedes (2011, p.30) “A utilização de diversos diagramas permite que falhas sejam descobertas, diminuindo a possibilidade de ocorrência de erros futuros”. Neste trabalho serão abordados apenas os diagramas de caso de uso, classe, sequência e atividade.

Figura 5. Como Realmente funciona um projeto.

Como realmente funciona um projeto



Fonte: Guedes (2011).

1.5.1. Diagrama Caso de uso

Primeiramente é importante saber a definição de casos de uso, que segundo Melo (2010, p. 57) “um caso de uso descreve uma sequência de ações que representam um cenário principal (perfeito) e cenários alternativos, com o objetivo de demonstrar o comportamento de um sistema (ou parte dele), através de interações com autores.” E ainda segundo Booch, Rumbaugh e Jacobson (2005, p. 231) “um caso de uso é uma descrição de umas sequências de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator”.

Para os autores Lima (2011) e Booch, Rumbaugh e Jacobson (2005) o diagrama de casos de uso tem um papel central na modelagem do comportamento de um sistema, representando toda a funcionalidade de um sistema, evidenciando interações externas com outras entidades. Os autores Booch, Rumbaugh e Jacobson (2005) definem “um diagrama de caso de uso como um diagrama que mostra um conjunto de casos de uso e atores e seus relacionamentos.”

1.5.2. Diagrama de Classes

O diagrama de classes é fundamental na modelagem de um Software, como demonstra Melo (2010), afirmando que ele é a estrela principal de um sistema orientado a objetos. Tal diagrama tem como principal finalidade detalhar as classes pertencentes ao modelo e identificar seus relacionamentos.

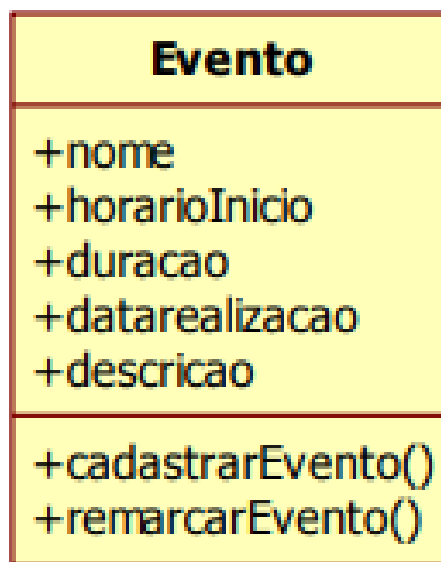
Segundo Booch, Rumbaugh e Jacobson (2005, p. 51) “classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.” Cada classe possui seus atributos e operações. Os atributos representam alguma propriedade da classe na qual ele esta inserido, uma classe não necessariamente precisa ter atributos. Já uma operação é uma atividade que pode ser realizada por um objeto da classe na qual esta operação esta inserida.

Como afirmam os autores Booch, Rumbaugh e Jacobson (2005) e Melo (2010), a maioria das classes não trabalham sozinhas, elas precisam se relacionar, colaborando uma com a outra. Os primeiros autores defendem a idéia de que existem três tipos de relacionamentos na modelagem orientada a objetos:

dependências, generalizações e associações. A segunda autora inclui ainda mais um relacionamento nesta lista, o de realização.

Um exemplo de uma classe pode ser visto na Figura 6 - Classe, a seguir:

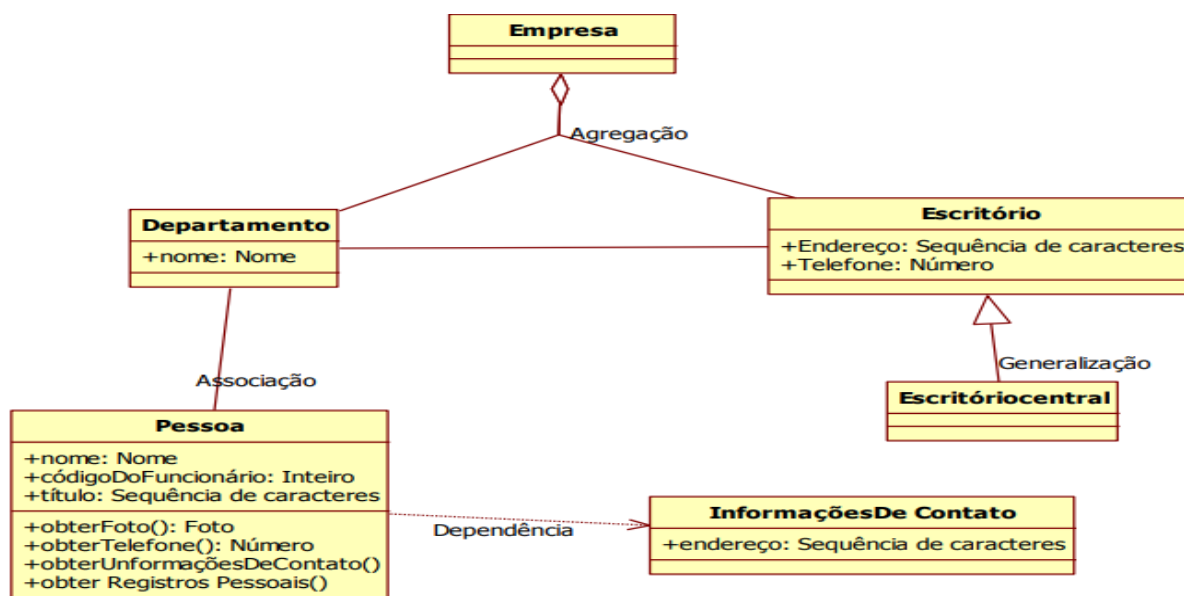
Figura 6. Classe



Fonte: Melo (2010, p.105).

E um exemplo de um diagrama de classe pode ser visto na figura 7 - Diagrama de classe a seguir:

Figura 7. Diagrama de classe



Fonte: Melo (2010, p.105).

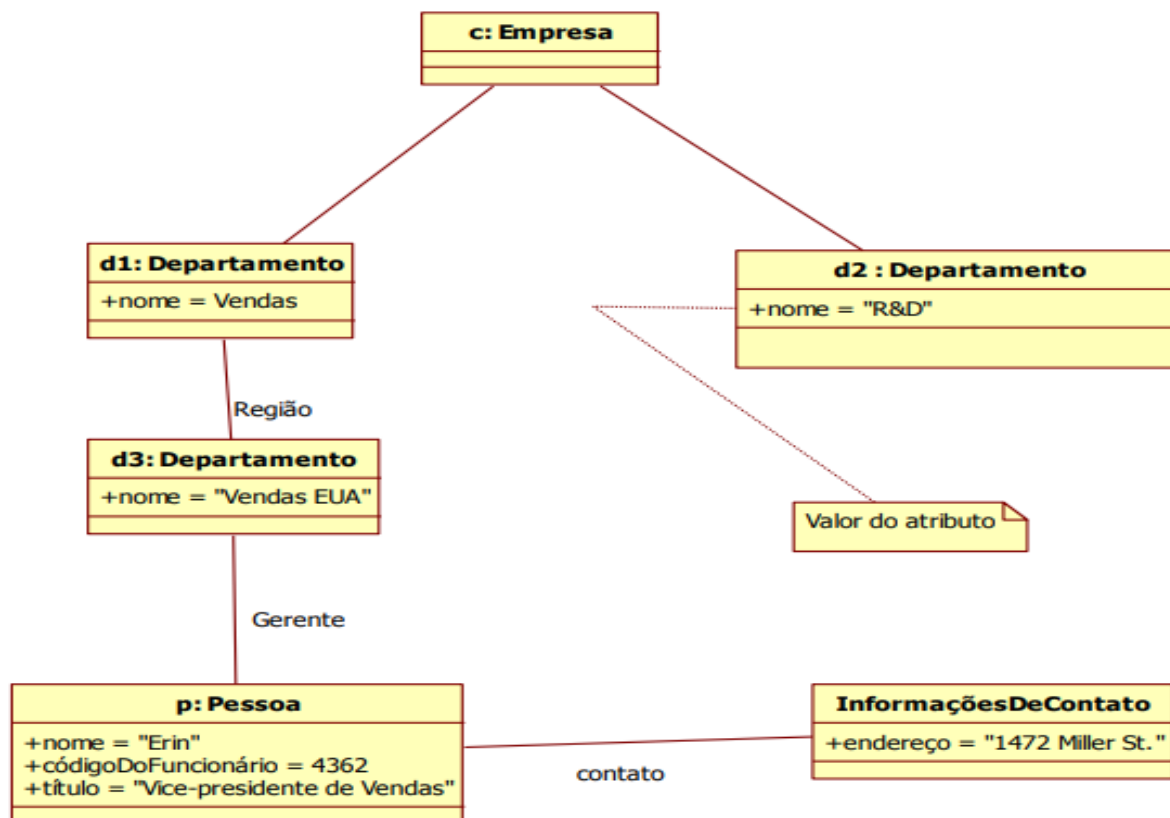
1.5.3. Diagrama de objetos

Os autores Booch, Rumbaugh e Jacobson (2005) e Melo (2010) afirmam que o diagrama de objetos representa uma instância de itens contidos em diagramas de classes, fazendo uma visão estática do projeto, modelando o sistema em determinado momento do processo do sistema.

Para Booch, Rumbaugh e Jacobson (2005), a definição de diagrama de objetos é “diagrama que mostra um conjunto de objetos e seus relacionamentos em um ponto do tempo. Graficamente, o diagrama de objetos é uma coleção de vértices e arcos”. Já para Melo (2010) “um diagrama de objetos consiste numa instância do diagrama de classes, no qual para cada classe tem-se um objeto (sua instância) em um determinado ponto do tempo”.

Tendo em vista a Figura 7 - Diagrama de Classe, pode ser visto a seguir, na Figura 8 - Diagrama de objetos, um exemplo de tal diagrama, representando uma instância de itens contidos em um diagrama de classe.

Figura 8. Diagrama de objetos



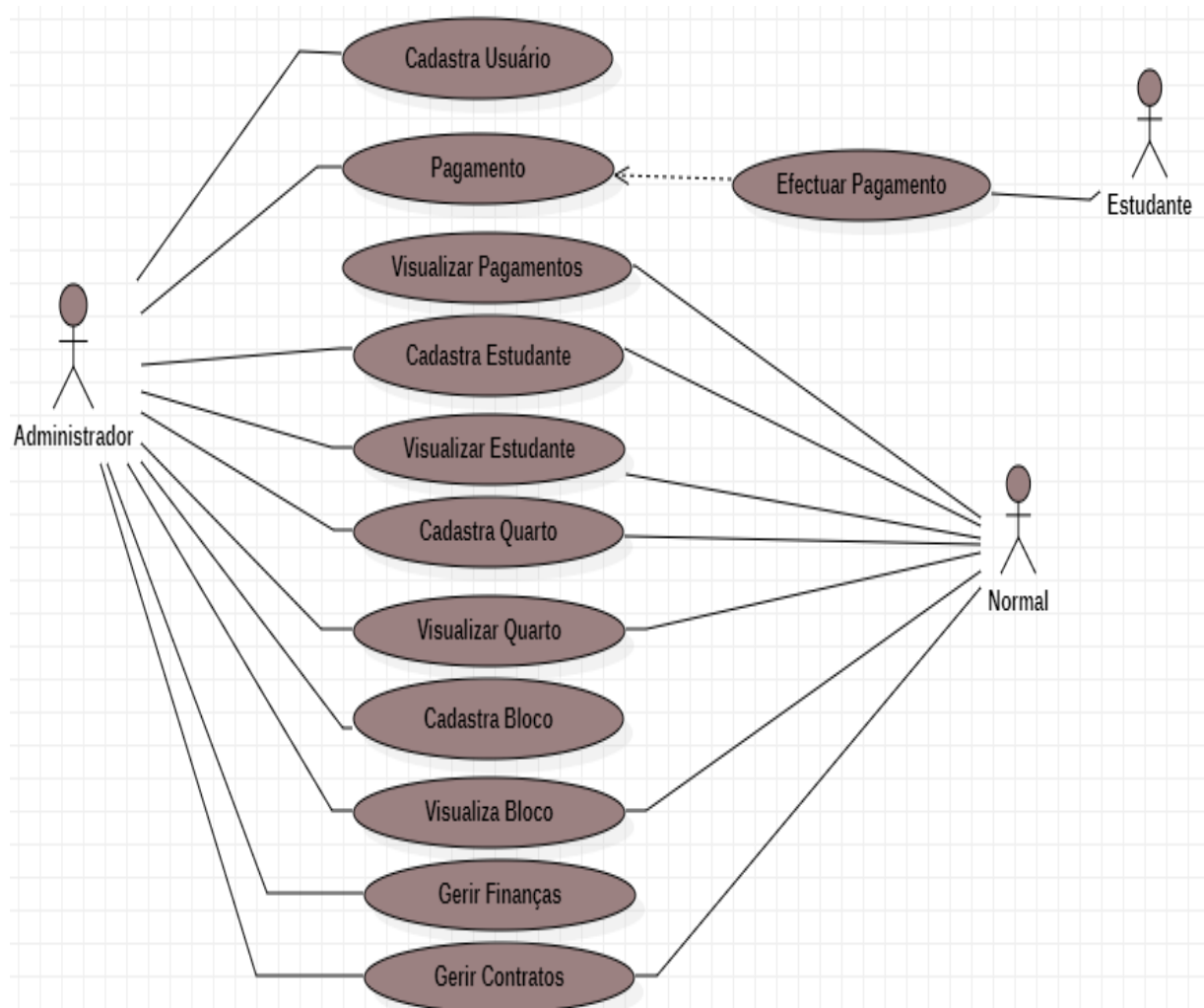
Fonte: Melo (2010).

1.5.4. Diagramas do sistema

1.5.4.1. Caso de Uso

Na Figura 9 serão apresentados os usuários do sistema com as suas funcionalidades. O software terá apenas dois usuários, o administrador que poderá acessar todas as funcionalidades e o Usuário Normal que terá um acesso limitado no sistema.

Figura 9. Caso de Uso

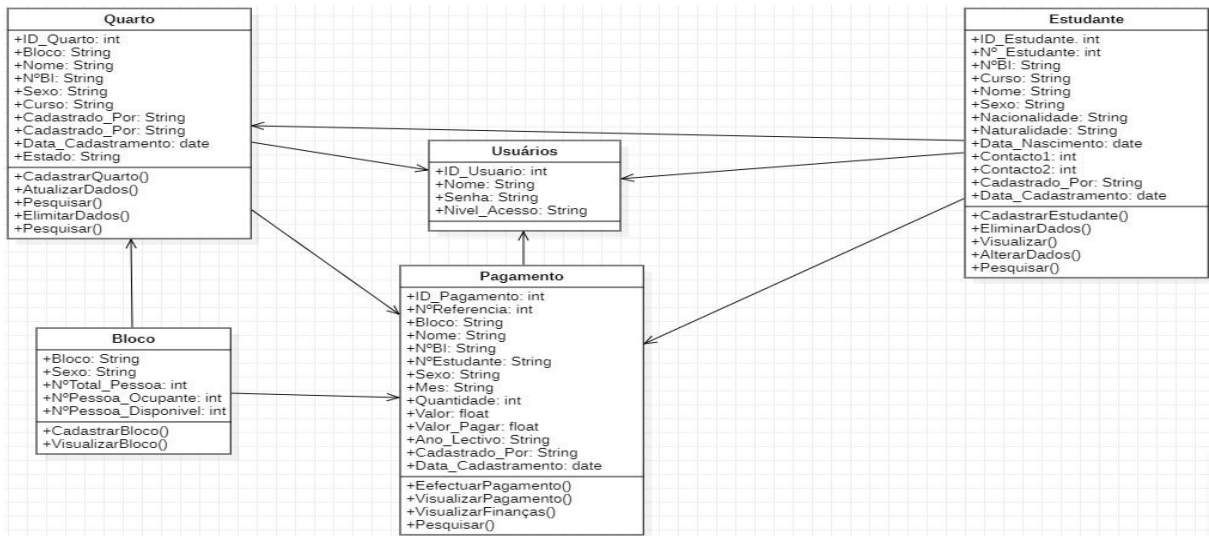


Fonte: Elaboração Própria.

1.5.4.2. Classe

Na Figura 10 serão apresentadas todas as classes do software com os seus atributos, métodos e os relacionamentos entre as classes.

Figura 10. Classe

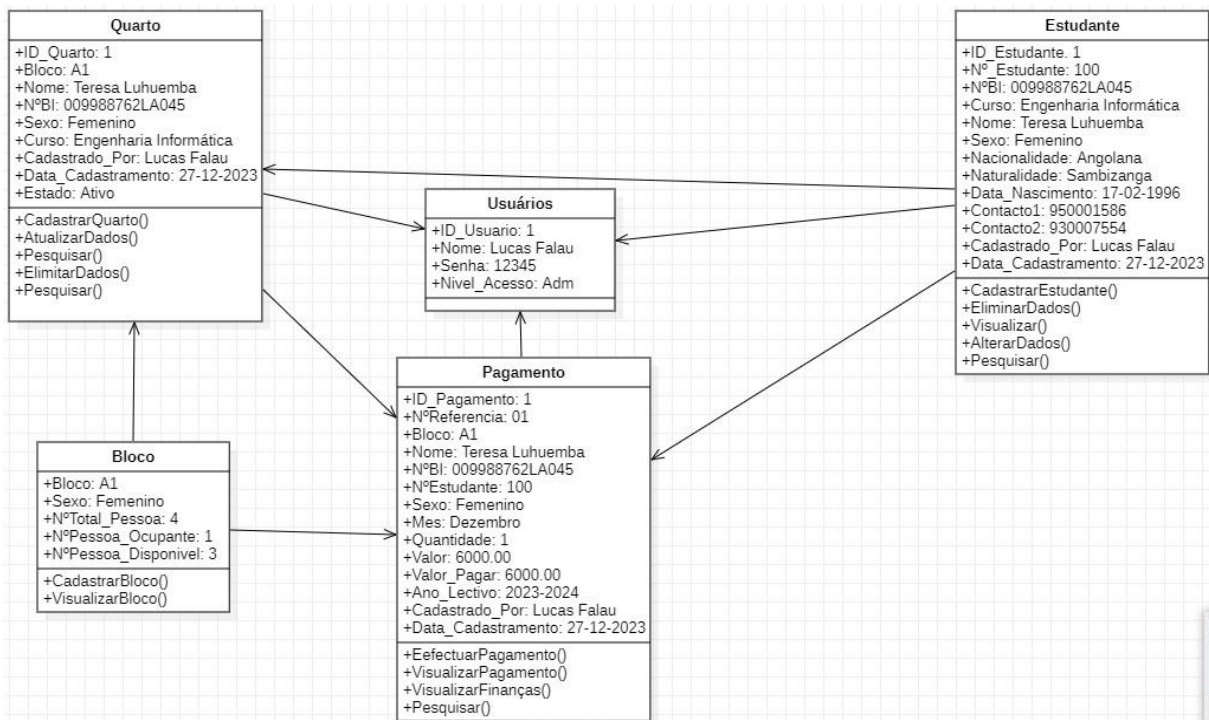


Fonte: Elaboração Própria.

1.5.4.3. Objecto

Na Figura 11 serão apresentados todos os objectos do software com os seus atributos, métodos e os relacionamentos entre as classes.

Figura 11. Objecto



Fonte: Elaboração Própria.

1.6. Banco de Dados

Banco de dados segundo Heuser (1998, p.3) é um conjunto de dados integrados que tem por objetivo atender a uma comunidade de usuários.

Elmasri e Navathe (2005, p.3) afirmam que os bancos de dados representam um papel crítico em quase todas as áreas em que os computadores são utilizados, incluindo negócios, comércio eletrônico, engenharia e ciências da informação. Os autores definem genericamente banco de dados como “uma coleção de dados relacionados, onde dados são fatos que podem ser gravados e que possuem significado implícito”. (Elmasri e Navathe, 2005, p.3).

Ainda segundo os mesmos autores o uso do termo banco de dados é geralmente mais restrito, e destacam-se as seguintes propriedades implícitas:

- Um banco de dados representa alguns aspectos do mundo real, sendo chamado, às vezes, de minimundo ou de universo de discurso (UoD). As mudanças no minimundo são refletidas em um banco de dados.
- Um banco de dados é uma coleção lógica e coerente de dados com algum significado inerente. Uma organização de dados ao acaso (randômica) não pode ser corretamente interpretada como um banco de dados.
- Um banco de dados é projetado, construído e povoado por dados, atendendo a uma proposta específica. Possui um grupo de usuários definido e algumas aplicações preconcebidas, de acordo com o interesse desse grupo de usuários.

1.6.1. Diagrama de entidade – Relacionamento (DER)

O conceito fundamental da abordagem ER é o conceito de entidade, que segundo Korth e Silberschatz (1995, p.7) é um “objeto que é distinguível de outro objeto por um conjunto específico de atributos.” Outro conceito importante é o de relacionamento, que segundo o mesmo autor é a associação entre várias entidades.

Cada entidade é representada por um conjunto de atributos, que são as propriedades que a definem.

Um relacionamento é uma associação entre diversas entidades. Já um conjunto de relacionamentos é uma coleção de relacionamentos do mesmo tipo.

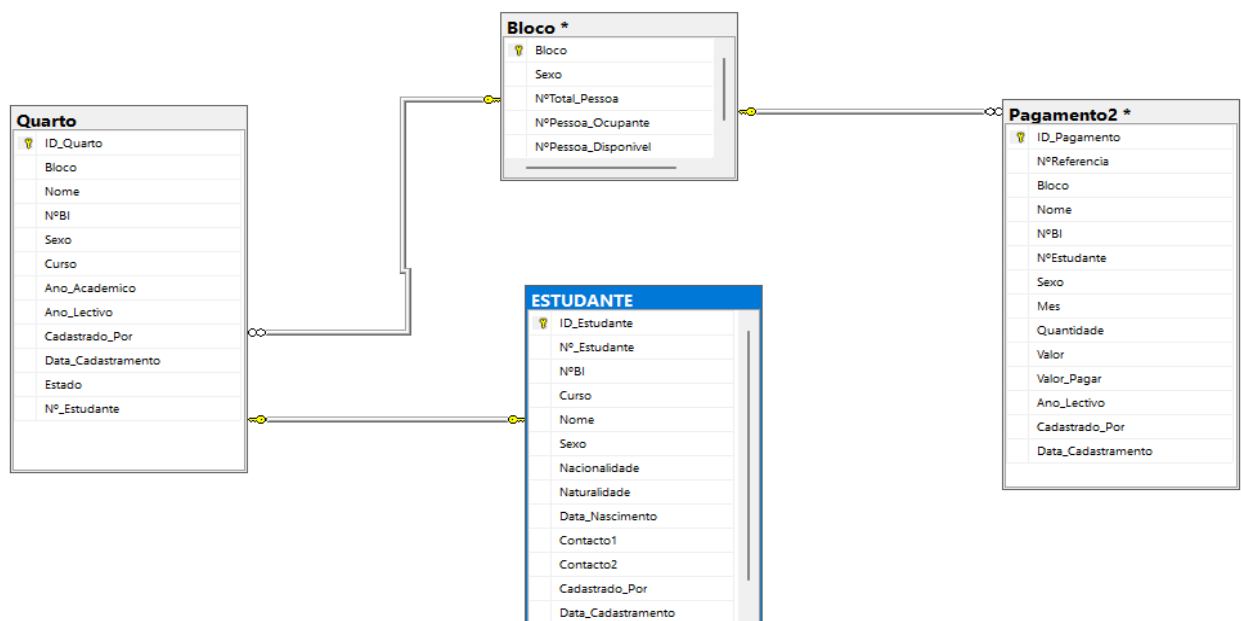
A estrutura lógica geral de um banco de dados pode ser expressa graficamente por um diagrama E-R, que consiste nos seguintes componentes:

- **Retângulos** – representam conjuntos de entidades;

- **Elipses** – representam atributos;
- **Losangos** – representam relacionamentos entre conjuntos de entidades;
- **Linhas** – ligam atributos a conjunto de entidades e conjunto de entidades a relacionamentos. Korth e Silberschatz (1995, p.8).

DER fornece uma visão lógica do banco de dados, fornecendo um conceito mais generalizado de como estão estruturados os dados de um sistema, na Figura 12. Será ilustrado o Diagrama de Entidade e Relacionamento do Sostware de Gestão de Apartamento desenvolvido.

Figura 12. Diagrama e de Relacionamento Entidade



Fonte: Elaboração Própria.

1.6.2. Sistemas de Gestão de Bases de Dados (SGBD)

Podemos armazenar dados de duas maneiras: utilizando bancos de dados, ou então, arquivos de dados permanentes (SILBERSCHATZ, KORF e SUDARSHAN e 2012). Com a utilização de um Banco de Dados temos seguintes vantagens:

- Controle centralizado de dados: os dados ficam localizados em um único local e isso facilita o controle e acesso.
- Controle da redundância: como os dados estão centralizados existe otimização e redução do espaço de armazenamento e controle de redundância.

- Compartilhamento de dados: com a utilização de um Banco de Dados sem redundância, o espaço de armazenamento é otimizado e facilita o compartilhamento de dados.

Segundo DATE (2004), um Sistema Gerenciador de Banco de Dados (SGBD) é um software genérico para manipular Banco de Dados. Ele permite a definição, construção e manejo de um Banco de Dados para diversas aplicações.

1.7. Projeto de Banco de Dados

Os sistemas de Banco de Dados tem um ciclo de vida para sua execução. ELMASRI e NAVATHE (2011) elencam as etapas em oito fases:

Tabela 4: Ciclo de vida do Banco de Dados

1. Definição do sistema: nesta etapa é determinado o escopo do sistema, o deverá ser armazenado, e quais serão as operações realização assim como seus usuários.
2. Projeto do Banco de Dados: esta etapa é a criação do projeto conceitual, lógico e físico.
3. Implementação do Banco de Dados: cria-se realmente o Banco de Dados, conforme esquemas definidos na etapa anterior.
4. Carga ou conversão de dados: nesta etapa, o Banco de Dados é preenchido com dados já existentes ou é preenchido manualmente.
5. Conversão de aplicação: nesta etapa os programas que antes acessavam o banco, são informados sobre as modificações do o novo banco (etapa auxiliar).
6. Teste e validação: verifica-se tudo está funcionando de acordo com o planejamento.
7. Operação: é relativo à disponibilização do sistema para uso;
8. Monitoramento e manutenção: nesta etapa observa-se o funcionamento do sistema para possíveis ajustes.

(Fonte: ELMASRI e NAVATHE (2011)).

1.8. SQL

A linguagem SQL é um padrão de linguagem de consulta de Banco de Dados que usa uma combinação de construtores em Álgebra e Cálculo Relacional e possui as seguintes características principais (ELMASRI e NAVATHE, 2005):

- Linguagem de definição de dados (DDL)
- Linguagem interativa de manipulação de dados (DML)
- Definição de Visões
- Autorização de usuários
- Integridade de dados
- Controle de Transações

Segundo Machado (2008), a primeira versão da linguagem SQL recebeu o nome de SEQUEL, entre 1976 e 1977, por razões jurídicas e depois de revisada, teve seu nome alterado para SQL. A IBM tinha um projeto grandioso chamado System R e foi logo colocado em execução e surgiram novas alterações na linguagem.

Houve um sucesso grande com a nova maneira para realização de consulta em banco, e a utilização da linguagem SQL aumentou. Dessa forma os SGBD's começaram a adquirir a linguagem como padrão DB2 da IBM, Oracle da Oracle Corporation, Sybase da Sybase Inc., Microsoft SQL Server da Microsoft, etc. Dessa forma a SQL tornou-se um padrão em bancos relacionais, faltava apenas tornar-se de direito. Em 1982, o American National Standard Institute (ANSI) tornou a SQL o padrão oficial de linguagem em ambiente relacional sendo hoje utilizada na grande maioria dos sistemas de Bancos de Dados relacionais, tais como MySQL, DB2, SQLServer.

Os comandos existentes na linguagem SQL são:

- **Comandos DDL (Data Definition Language):** Conjunto de comandos responsáveis pela criação, alteração e deleção da estrutura das tabelas e índices de um sistema (DATE, 2004).
- **Comandos DML (Data Manipulation Language):** Conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em um banco de dados (HEUSER, 2004).

❖ Linguagem de definição e manipulação de dados

Basicamente uma linguagem de definição de dados (DDL) é usada para definir o esquema conceitual do Banco de Dados. Na maioria dos SGBDs, a DDL também define as visões dos usuários, e algumas vezes as estruturas de armazenamento já uma linguagem de manipulação de dados (DML) é empregada para especificar as recuperações e atualizações do Banco de Dados (GARCIA-MOLINA, ULLMAN e WIDOM, 2008).

Os SGBDs também facilitam a conversão e a reorganização do Banco de Dados. Dessa forma podemos dizer que os SGBDs são programas de computador, que ajudam na:

- Definição e construção do Banco de Dados: que é o processo de criar uma estrutura inicial com tabelas e preenche-las com dados;
- Manipulação dos dados do Banco de Dados: são as operações de consultas alteração, exclusão realizadas nos dados.

1.9. Linguagem de Programação C#

C# é uma linguagem de programação, orientada a objetos que permite que os desenvolvedores utilizem os conceitos de encapsulamento, herança e polimorfismo, tornando o projeto mais seguro e robusto (PRENTICE, 2010).

A linguagem C# foi escolhida para o desenvolvimento desse trabalho, devido ao fato de permitir recursos de programação orientada a objetos, a facilidade de tratar erros e por se ter maior domínio em relação a ela.

1.10. Telas e Designer do Software

A seguir serão demonstradas as principais telas do TFC Software, isso irá facilitar para que o usuário compreenda as funcionalidades do sistema.

1.10.1. Tela de Processamento

A tela de processamento é a primeira tela do sistema, conforme demonstra a figura 13.

Figura 13. Tela de processamento



Fonte: Elaboração Própria.

1.10.2. Tela de login

A segunda tela do sistema é a tela de login, na qual o funcionário terá que informar os dados de acesso ao sistema, após o que, o sistema ira automaticamente identificar o usuário. Existem dois níveis de acesso, o Administrador que terá acesso total ao sistema e Normal com acesso limitado, o que se pode observar na Figura 14.

Figura 14. Tela de Login

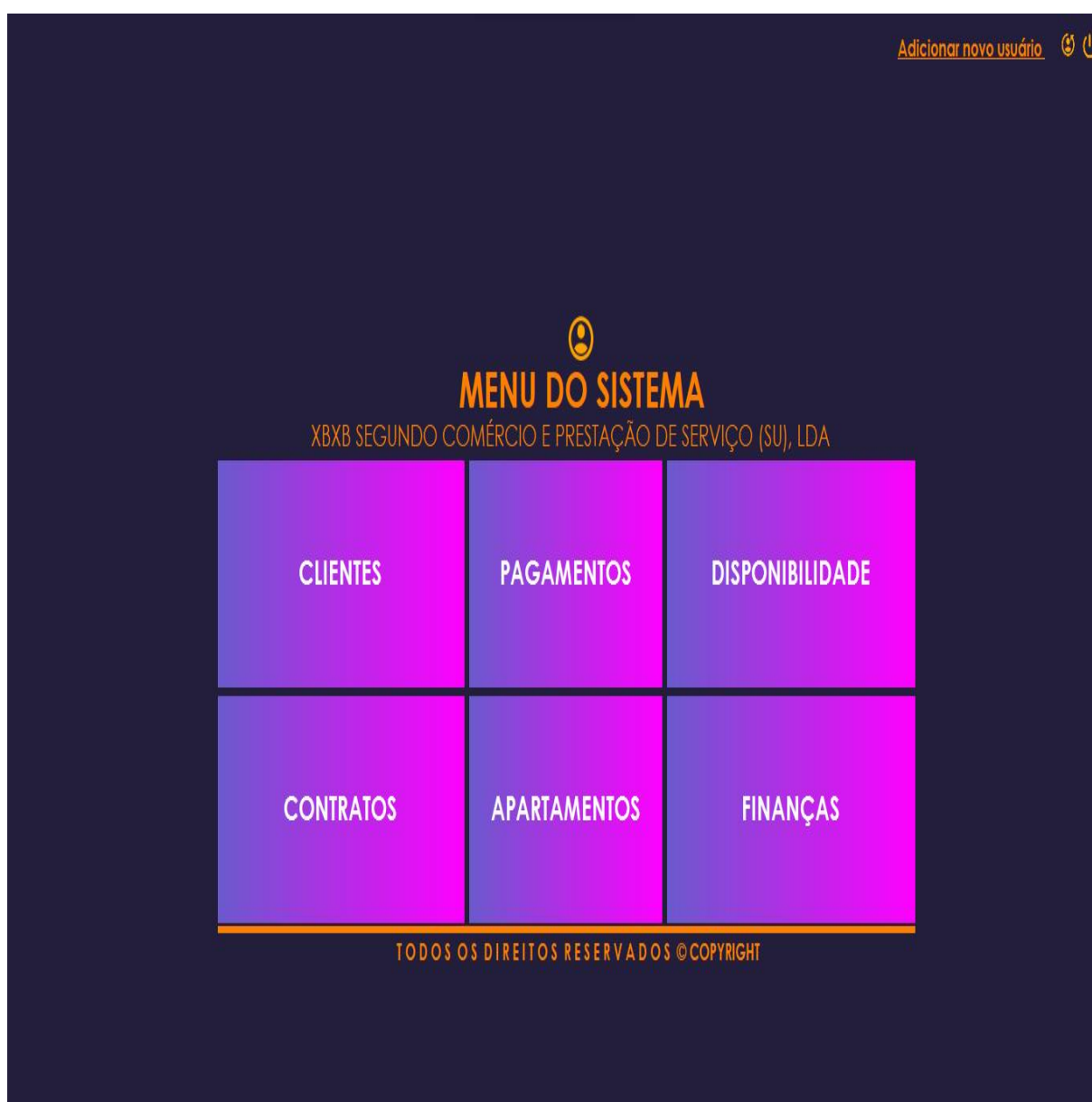
A screenshot of a login screen with a white background. It features three input fields: "USUÁRIO", "SENHA", and "NÍVEL DE ACESSO" (with a dropdown arrow). Below the fields is a horizontal orange line, followed by the text "TODOS OS DIREITOS FORAM RESERVADOS ©COPYRIGHT". At the bottom are two buttons: "ENTRAR" (orange) and "CANCELAR" (blue).

Fonte: Elaboração Própria.

1.10.3. Tela de menu do sistema

Após efetuar um login em um dos níveis de acesso, será aberta a tela de menu, que permitirá ao administrador ter acesso a todas as funcionalidades do sistema, sem nenhuma restrição, já o Normal não terá acesso às opções, Cadastrar Bloco, Finanças e Adicionar novos usuários. Nessa tela será possível acessar os formulários através dos botões com as descrições dos formulários, o que demonstra a Figura 15.

Figura 15. Menu do Sistema.



Fonte: Elaboração Própria.

1.10.4. Tela de cadastro de Usuários

Nessa tela onde apenas o administrador tem acesso é possível cadastrar, visualizar e excluir um usuário. O sistema não permitirá que seja eliminado um usuário se o campo “ID” estiver em branco, conforme a Figura 16.

Figura 16. Cadastramento de Usuário

Adicionar novo usuário

DADOS DO USUÁRIO

INSIRA O ID EM CASO DE ELIMINAR O USUÁRIO NOME DO USUÁRIO SENHA

LISTA DOS USUÁRIO

ID	USUÁRIO	NÍVEL DE ACESSO	SENHA
1	lukasfalau.com	Administrador	12345
2	lukascundende.com	Normal	12345

GRAVAR ELIMINAR

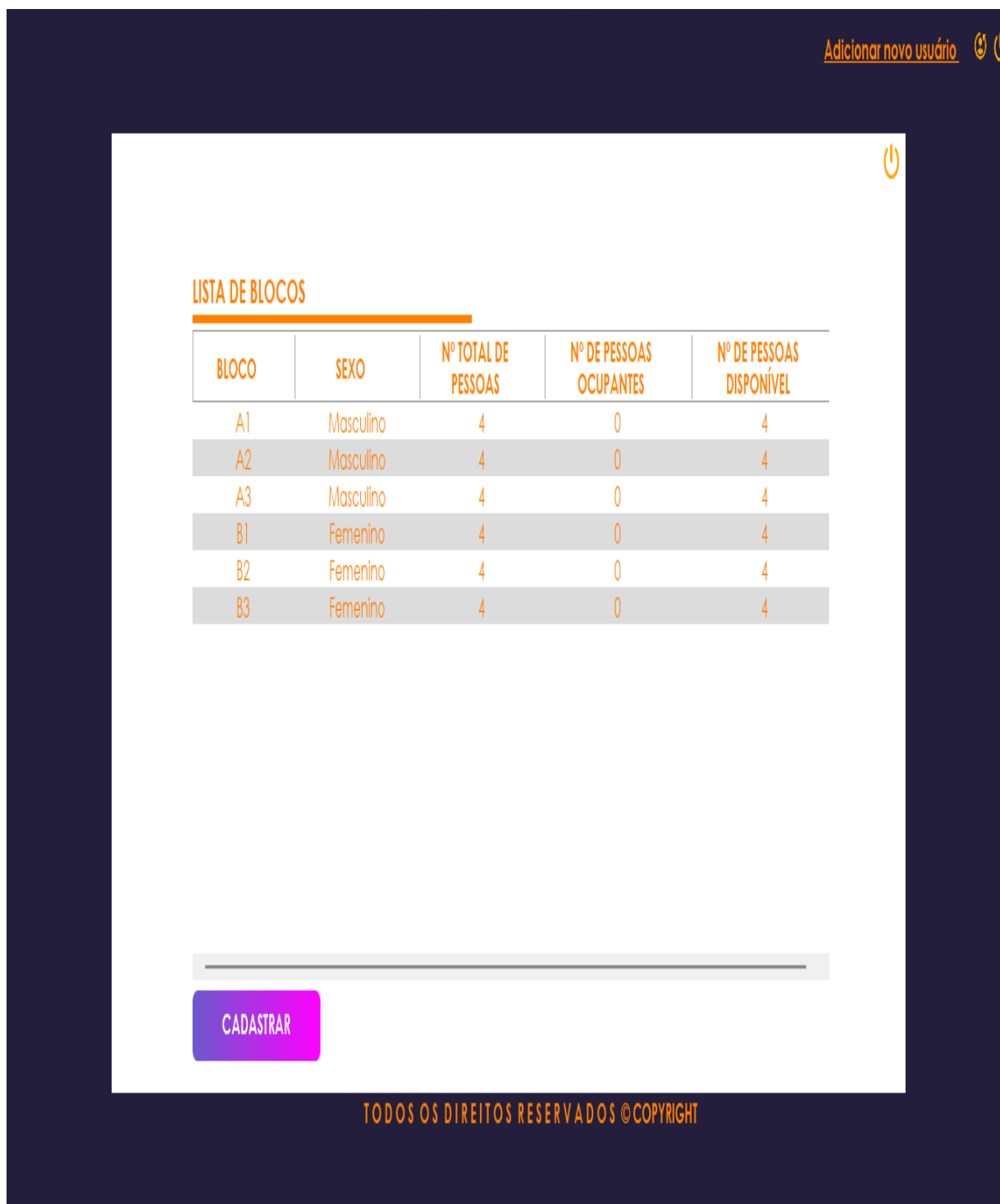
TODOS OS DIREITOS RESERVADOS ©COPYRIGHT

Fonte: Elaboração Própria.

1.10.5. Tela de Disponibilidade

Nessa tela, faremos a verificação da disponibilidade de vagas existente nos nossos apartamentos.

Figura 17. Tela de Disponibilidade.



Fonte: Elaboração Própria.

1.10.6. Tela de Cadastramento de Clientes

Para o cadastramento de Clientes, o sistema permite salvar, eliminar, editar e visualizar, conforme mostra a figura 18.

Figura 18. Tela de Cadastramento de Cliente

The screenshot displays a web application interface for client registration. At the top right, there is a link "Adicionar novo usuário" and two user profile icons. The main content area is titled "DADOS DO CLIENTE" and contains a form with the following fields:

ID	NACIONALIDADE
NOME DO CLIENTE	NATURALIDADE
Nº DE IDENTIFICAÇÃO	CONTACTO 1
Nº DO BILHETE DE IDENTIDADE	CONTACTO 2
PROFISSÃO	lukasfalau.com
SEXO	06/01/2024 ▼

Below the form, there are four action buttons: "SALVAR", "ELIMINAR", "EDITAR", and "VISUALIZAR". At the bottom of the page, a copyright notice reads "TODOS OS DIREITOS RESERVADOS ©COPYRIGHT".

Fonte: Elaboração Própria.

1.10.7. Tela de Gestão de Contrato

Nessa tela, será possível localizar os dados do cliente, renovar, terminar e visualizar contratos, conforme exibe a Figura 19.

Figura 19. Tela de Contratos

Adicionar novo usuário

DADOS DO CONTRATO

INSIRA O ID DO APARTAMENTO A SER PESQUISADO

CLIENTE MODALIDADE

PROFISSÃO Nº DO BILHETE

LOCALIZAR RENOVAR CONTRATO TERMINAR CONTRATO VISUALIZAR

TODOS OS DIREITOS RESERVADOS ©COPYRIGHT

Fonte: Elaboração Própria.

1.10.8. Tela de Finanças

Nessa tela, calculamos a renda pelo ano ou de uma forma geral, de acordo a Figura 20.

Figura 20. Tela de Finança

Adicionar novo usuário

Nº DA REFERÊNCIA	VALOR DA RENDA	ANO
------------------	----------------	-----

VALORTOTAL: 0

LOCALIZAR CALCULAR INSIRA O ANO LECTIVO

TODOS OS DIREITOS RESERVADOS ©COPYRIGHT

Fonte: Elaboração Própria.

1.10.9. Tela de Pagamento de Renda

Para efectuar o pagamento, basta apenas localizar o cliente, pesquisando pelo Nº do Cliente e os dados do mesmo serem exibidos nos campos correspondentes ao Estudante, adicionar o valor, quantidade dos meses para permitir calcular o valor total que o cliente deverá pagar, na Figura 21. Pode-se ver como esse processo funciona.

Figura 21. Tela de Pagamento de Renda.

Adicionar novo usuário

DADOS DO PAGAMENTO

INSIRA O Nº DO CLIENTE A SER PESQUISADO

MODALIDADE

CLIENTE lukasfalau.com

Nº DO BILHETE VALOR

MES À PAGAR VALOR PAGAR

Nº DA REFERÊNCIA ANO

APARTAMENTO QUANTIDADE

LOCALIZAR CALCULAR PAGAMENTO VISUALIZAR

TODOS OS DIREITOS RESERVADOS ©COPYRIGHT

Fonte: Elaboração Própria.

CAPÍTULO II. METODOLOGIA DE PESQUISA

2.1. Tipos de Pesquisa

Para este trabalho, usaremos a pesquisa Descritiva, que a sua definição é:

Segundo Gil (1999), as pesquisas descritivas têm como finalidade principal a descrição das características de determinada população ou fenômeno, ou o estabelecimento de relações entre variáveis. São inúmeros os estudos que podem ser classificados sob este título e uma de suas características mais significativas aparece na utilização de técnicas padronizadas de coleta de dados.

A pesquisa descritiva expõe as características de determinada população ou fenômeno, estabelece correlações entre variáveis e define sua natureza. "Não têm o compromisso de explicar os fenômenos que descreve, embora sirva de base para tal explicação" (Vergara, 2000, p. 47).

2.2. Modelos de Abordagem

Para o modelo de abordagem, usaremos a pesquisa quali – quantitativa:

"A pesquisa quali-quantitativa é aquela que utiliza simultaneamente elementos da pesquisa qualitativa e quantitativa, sendo que seu objetivo principal é coletar dados de diferentes naturezas para a análise e interpretação dos resultados" (Bardin, 2016, p. 82).

"A pesquisa quali-quantitativa se caracteriza pela utilização de diferentes estratégias metodológicas e técnicas de coleta de dados, como questionários, entrevistas, observação, entre outras, com o objetivo de obter informações detalhadas e amplas sobre um fenômeno ou problema em estudo" (Marconi & Lakatos, 2017, p. 223).

2.3. Nível de estruturação de método

a) Nível teórico

O nível teórico deste trabalho inclui a revisão bibliográfica sobre os conceitos e técnicas relevantes para o desenvolvimento de software e a gestão de apartamentos. Serão explorados conceitos de engenharia de software, como metodologias ágeis, modelagem de processos e testes de software, bem como

conceitos de gestão de apartamentos, como administração de condomínios, gestão de contratos de aluguel e venda, e gestão de inadimplência.

b) Nível empírico

O nível empírico do presente trabalho envolve a realização de uma pesquisa descritiva com abordagem qualitativa e quantitativa, que tem como objetivo coletar dados sobre a gestão de apartamentos pela empresa e avaliar a efetividade do software desenvolvido para a gestão desses apartamentos.

c) Nível Estatístico

O nível estatístico será utilizado para a análise quantitativa dos dados coletados durante a pesquisa empírica. Serão aplicados métodos estatísticos para avaliar os resultados obtidos, incluindo a utilização de medidas de tendência central, como a média e a mediana, e medidas de dispersão, como o desvio padrão e a variância.

2.4. Técnicas de recolha de dados

Pesquisa bibliográfica: será realizada uma revisão sistemática da literatura para identificar as principais teorias, conceitos e métodos relacionados ao desenvolvimento de software e à gestão de imóveis. Serão utilizados livros, artigos científicos, revistas especializadas, entre outros.

Estudo de caso: será realizado um estudo de caso para avaliar a efetividade do software desenvolvido para a gestão de apartamentos. Serão coletados dados sobre a utilização do software pelos funcionários da empresa, bem como as melhorias e dificuldades encontradas no processo de gestão de apartamentos.

Entrevistas: serão realizadas entrevistas com os funcionários da empresa para coletar dados sobre a utilização do software e as melhorias que podem ser feitas na solução implementada. As entrevistas serão realizadas de forma individual e gravadas para posterior análise.

Questionários: será aplicado um questionário aos clientes da empresa para avaliar sua satisfação com o processo de gestão de apartamentos e a utilização do software desenvolvido. O questionário será elaborado com base em escala Likert e permitirá uma análise quantitativa dos dados coletados.

2.5. Instrumentos e procedimentos de recolha de dados

No presente trabalho usaremos os seguintes instrumentos:

- a. Roteiros de entrevistas
- b. Guias de observação

2.6. População e amostra

Forão selecionados 15 trabalhadores que equivale a 100% da população para participar das entrevistas, de diversas áreas de forma a obter informações sobre o processo de gestão de imóveis e a utilização do software a ser desenvolvido. A seleção dos funcionários foi baseada na disponibilidade e interesse em participar da pesquisa.

2.7. Dificuldades encontradas

Ao longo da elaboração do nosso trabalho encontramos dificuldades na recolha de dados, mais que depois foram superados.

CAPÍTULO III. DISCUSSÃO E RESULTADOS

3.1. Resultados da Implementação

➤ Interface do Usuário

A interface do usuário foi desenvolvida seguindo as melhores práticas de designer de UI. Foram aplicados princípios de usabilidade e acessibilidade para garantir uma experiência de usuário otimizada e intuitiva.

➤ Funcionalidades Implementadas

Foram implementadas funcionalidades abrangentes que abordam todas as etapas do processo de gestão de apartamentos, desde o cadastro inicial até a administração contínua. Isso inclui recursos robustos para gerenciamento de propriedades, contratos de aluguel, pagamentos e relatório de finança.

➤ Integração do Software

O software foi integrado de maneira eficiente com o software de Gestão de Bases de Dados SQL Server e a Linguagem de Programação C#.

3.2. Recursos Necessários para o Desenvolvimento do Projeto

Em seguida mostraremos os recursos que foram necessários para o desenvolvimento do nosso projeto:

- 1 (Um) Programador;
- 1 (Um) Computador;
- 1 (uma) Impressora;
- IDE para o desenvolvimento: Visual Studio;
- IDE para modelagem de dados: Azure Data Studio;
- Linguagem de Programação: C# (Csharp);
- Banco de Dados; Sql Server

3.3. Estimativa de Custos

Os custos estimados para a concepção do nosso projeto foram:

Programador

- Custo: 50.000.00kz;

Computador

- Custo: 250.001.70kz;

Impressora

- Custo: 59.982.00kz;

IDE para o desenvolvimento: Visual Studio;

- Custo: Grátis

IDE para modelagem de dados: Azure Data Studio;

- Custo: Grátis

Linguagem de Programação: C# (Csharp);

- Custo: Grátis

Banco de Dados: Sql Server.

- **Custos total: 359 983.70kz**

3.4. Orçamento do Projeto

Podemos observar na tabela 5 o orçamento do nosso projeto.

Tabela 5. Orçamento do Projeto

Recursos	Valor
Programador	50.000.00kz
Computador	250.001.70kz
Impressora	59.982.00kz
IDE para o desenvolvimento: Visual Studio	0,00kz
IDE para modelagem de dados: Azure Data Studio	0,00kz
Linguagem de Programação: C# (Csharp)	0,00kz
Banco de Dados: Sql Server.	0,00kz
Custo Total do Projeto	359 983.70kz

Fonte: Elaboração Própria.

3.5. Discussão dos resultados

Os resultados obtidos estão alinhados com o objetivo principal do projeto, que era desenvolver um software de gestão de apartamentos com o intuito de automatizar todos os processos envolvidos na gestão de apartamentos desde o cadastro até a administração. O software atende às expectativas e requisitos estabelecidos, proporcionando uma solução eficiente e personalizada.

A implementação do software vai resultar em melhoria significativas na eficiência operacional da empresa XBXB Segundo Comércio Geral e Prestação de Serviço (SU). Os processos manuais serão substituídos por fluxos de trabalho automatizados, reduzindo o tempo necessário para realizar tarefas administrativas e aumentar a precisão dos dados. A interface do usuário vai contribuir para uma experiência positiva do usuário, aumentando a produtividade e reduzindo a curva de aprendizado.

Embora o software atenda às necessidades atuais da empresa XBXB Segundo Comércio Geral e Prestação de Serviço (SU), reconhecemos que sempre há espaço para melhorias contínuas. Planeamos continuar monitorando o desempenho do software e aprimoramento, incluindo novas funcionalidades, melhorias na interface do usuário e integração com tecnologias emergentes.

CONCLUSÃO

A fim de uma longa jornada de pesquisa, estando em conformidade com as normas e métodos do Instituto Superior de Angola, tendo em conta que as nossas hipóteses foram alcançadas, chegamos as seguintes conclusões:

Foi possível constatar que os objetivos estabelecidos no nosso trabalho foram alcançados. A concepção e implementação do sistema proporcionaram uma solução robusta e eficiente para atender às necessidades específicas da gestão de apartamentos, promovendo maior organização e otimização dos processos relacionados.

A gestão de imóveis é um mercado muito competitivo e cada vez mais surgem as necessidades de automatizar todos os processos relacionadas à gestão de apartamentos para proporcionar um ambiente de trabalho eficiente, gerando resultados e satisfação dos clientes.

Com o software para gestão a empresa XBXB Segundo Comercio e Prestação de Serviço (SU), LDA vai garantir segurança e facilidade nos dados que a empresa lida na gestão de seus apartamentos, garantindo um ambiente moderno usando métodos tecnologicos para o tratamento das informações.

SUGESTÕES

Sugere-se garantir a centralização de dados e o acesso remoto ao software. Isso permitirá que os responsáveis pela gestão tenham acesso instantâneo a informações cruciais, independentemente dos diversos sectores da empresa, contribuindo para uma resposta ágil a demandas e emergências.

Capacitar os trabalhadores concernentes à utilização de meios tecnológico, a fim de facilitar a utilização do software para gestão de seus apontamentos.

Recomenda-se estabelecer um feedback contínuo, envolvendo os usuários finais e a equipe de desenvolvimento. Essa abordagem permitirá identificar oportunidades de melhoria à medida que o software é utilizado na prática.

REFERÊNCIAS BIBLIOGRÁFICAS

- ✓ Baptista, C. Sousa, M. (2011). Como fazer Investigação. (4ª ed). Lisboa: Pactor.
- ✓ Bardin, L. (2016). Análise de conteúdo. São Paulo. SP: Edições 70.
- ✓ Bresser-Pereira, L. C. (2004). Políticas de Gestão do Estado e Políticas Econômicas: Novos Desafios. Revista de Economia Política, 24(4), 26-41.
- ✓ Booch, G.; James, R.; Ivar, J. (2005). UML: Guia do usuário. 6ª Reimpressão. Rio de Janeiro: Elsevier.
- ✓ Birkinshaw, J. Goddard, J. (2009). O que é Implementação? Organization Science, 20(1), 311-320.
- ✓ BROWN, N. (2010). Managing technical debt in software-reliant systems. In: . [S.l.: s.n.].
- ✓ Chiavenato, I. (2000). Introdução à teoria geral da administração. 6. ed. Rio de Janeiro: Campus.
- ✓ Cervo, A. L. Bervian, P. A. (2002). Metodologia científica. 5. ed. São Paulo: Prentice, H. A. (2010). Linguagem de Programação C#. Rio de Janeiro.
- ✓ DROBKA, J.; NOFTZ, D.; RAGHU, R. (2004). Piloting xp on four mission-critical projects. IEEE Software.
- ✓ Elmasri, R.; Navathe, S. B. (2005). Sistemas de Bancos de Dados. 4. ed. Edição em português. Addison-Wesley.
- ✓ ELMASRI, R.; NAVATHE, S. B. (2011). Sistemas de Bancos de Dados. 6. ed. Addison-Wesley.
- ✓ Fortin, M. (2009). O Processo de Investigação: Da concepção à realização. (5 ed).
- ✓ Gil, A. C. (1999). *Métodos e técnicas de pesquisa social*. 5. ed. São Paulo: Atlas.
- ✓ Guedes, G. T. A. (2011). UML: Uma abordagem prática. 2º edição. São Paulo: Novatec.
- ✓ Heuser, Carlos A. (2001). Projeto de banco de dados. Porto Alegre: Sagra Luzzatto. 203p.
- ✓ Heuser, C. A. (2004). Projeto de Banco de Dados, 5a. ed. Sagra Luzzato.
- ✓ Hill, M. Hill, A. (2002). Investigação por Questionário. (2 ed). Lisboa: Edições Sílabo.

- ✓ Korth, Henry F., silberchatz, Abraham; Sudarshan, S. (1995). Sistema de Banco de Dados. São Paulo: Pearson Makron Books.
- ✓ Lima, A. S. (2011). UML Do requisito à solução. 1º edição. São Paulo: Érica.
- ✓ MACHADO, F.N.R. (2008). Projeto de Banco de Dados: uma visão prática. 15. ed. São Paulo: Érica.
- ✓ Marconi, M. A; Lakatos, E. M. (2017). Fundamentos de metodologia científica. 8ed. São Paulo: Atlas.
- ✓ Melo, A.C. (2010). Desenvolvendo aplicações com UML do conceito à implementação. 3ª edição. Rio de Janeiro: Brasport.
- ✓ Ostrom, V. (1976). Uma Abordagem Comportamental para a Teoria da Escolha Racional da Ação Coletiva. The American Political Science Review, 92(1), 1-22.
- ✓ Paula, F. W, P. (2001). Engenharia de Software: fundamentos, métodos e padrões. 2. ed. Rio de Janeiro: LTC.
- ✓ Pires, A. R. (2007), "Qualidade – Sistemas de Gestão da Qualidade, 3. ed. Edições Sílabo: Lisboa.
- ✓ Pfleeger, Shari Lawrence. (2004). Engenharia de Software: teoria e prática. 2ª Edição. São Paulo: Prentice Hall.
- ✓ Pressman, R. S. (2011). Software engineering: a practitioner's approach. (7th ed). New York: McGraw-Hill.
- ✓ Pressman, R. S. (2016). Engenharia de Software: Uma abordagem Prática. 8. ed. New York: McGraw-Hill.
- ✓ Paula, F. Wilson, P. (2003). Engenharia de Software. Rio de Janeiro: LTC.
- ✓ Pressman, R. S. (2006). Engenharia de Software. 6. ed. Rio de Janeiro: McGraw-Hill.
- ✓ Santos, A. (2008). "Gestão Estratégica – Conceitos, Modelos e Instrumentos", Escolar. Editora: Lisboa.
- ✓ Sommerville, I. (2007). Engenharia de Software. 8. ed. England: Addison Wesley.
- ✓ SOMMERVILLE, I. (2010). Software Engineering. 9. ed. Harlow. England: Addison-Wesley.
- ✓ Stair, R. M.; Reynolds, G. W. (2011). Princípios de sistemas de informação. 9. ed. São Paulo: Cengage Learning.

- ✓ Tonsig, S. L. (2008). Engenharia de Software-Análise e Projeto de Sistemas. 2ª edição. Rio de janeiro: Ciência Moderna Ltda.
- ✓ Turban, Efraim; Jr., R. Kelly Rainer; Potter, Richard E. Introdução a Sistemas de Informação: Uma Abordagem Gerencial. 4ª Edição. Rio de Janeiro: Elsevier, 2007.
- ✓ Vergara, S. C. (2000). Projetos e relatórios de pesquisa em administração. 3. ed. Rio de Janeiro: Atlas.

APÊNDICES



ZICAI GRUPO, LDA
NIF: 5417012670
Vila de Cacuaco
Cacuaco - Luanda
Telef. 929050530
E-mail: info@zicaigrupo.com

Pro-Forma

FACTURA PROFORMA

Data

2024-04-30

Número

PP 20012024/956

LUCAS
LUANDA
LUANDA
Angola

ORIGINAL

Página 01

Entidade	V/Contribuinte	Vendedor	V/Encomenda	Mod.Pag.	Armazém	Validade
999999999	-----			PRONTO P	1	2024-04-30

ESTA PROFORMA TEM VALIDADE DE 2 DIAS

Cód. Produto	Designação	Un	Quant.	Unitário	Desc.	Total
ALPYT00E115K	PORT TOSHIBA DYNABOOK 11.6 HD N4020 4GB	UN	1,00	250 001.70		250 001.70
5AR83B	IMPRESSORA HP DESKJET 2710	UN	1,00	59 982.00		59 982.00



Sub-Total	Incidência IVA	Taxa	Total IVA	Total Desc.	Total
309 983.70	309 983.70	14%	43 397.72	0.00	309 983.70
		%		IVA CATIVO:	0.00
			TOTAL DO IVA:		43 397.72
			TOTAL DO DOCUMENTO + IVA:		353 381.42

Este documento não serve de factura

Extensão: Trezentos e Cinquenta e Tres Mil Trezentos e Oitenta e Um KWANZAS e Quarenta Dois CENTIMOS

BIR - Akz: 001463817-10-001 - IBAN: AO06 0067 0000 0146 3817 1018 9
BAI - Akz: 217555412-10-001 - IBAN: AO06 0040 0000 1755 5412 1016 0
BFA - Akz: 38934302-30-001 - IBAN: AO06 0006 0000 3893 4302 3012 4
BIC - Akz: 042789478-10-001 - IBAN: AO06 0051 0000 4278 9478 1013 7
SOL - Akz: 83711323-10-001 - IBAN: AO06 0044 0000 8371 1323 1019 3
ATL - Akz: 29295864-10-001 - IBAN: AO06 0055 0000 2929 5864 1018 3

Hora: 14:34

Zuke-Processado por programa validado n.29/AGT/2019 - Par-Software
Licenciado a ZICAI GRUPO, LDA

Operador: MARIA DE FÁTIMA



ANEXOS



INSTITUTO
SUPERIOR
DE ANGOLA

À
DIRECÇÃO DA EMPRESA XBXB SEGUNDO
COMÉRCIO E PRESTAÇÃO DE SERVIÇO

LUANDA

CREDENCIAL

Ref.: 388/DAC/ISA/023

Melhores Cumprimentos,

Excelência,

O Instituto Superior de Angola – ISA, sua criação foi aprovada pelo Decreto Presidencial nº 168/12 de 24 de Julho, possui a sede em Luanda, bairro da Pedreira, município de Cacucaco, e aposta na formação de quadros qualificados e na investigação científica para dar respostas as necessidades locais e nacionais de acordo com as orientações emanadas pelo Executivo Central.

Havendo a necessidade de credenciar e enviar nos diferentes locais para pesquisar a temática do Trabalho de Fim de Curso para obtenção do grau de Licenciatura em **ENGENHARIA INFORMÁTICA** o (a) estudante:

NOME: **LUCAS FALAU CUNDENDE**

TEMA: **CONCEPÇÃO E IMPLEMENTAÇÃO DE UM SOFTWARE DE GESTÃO DE APARTAMENTOS NA EMPRESA XBXB SEGUNDO COMÉRCIO E PRESTAÇÃO DE SERVIÇO (SU), LDA.**

Vimos por intermédio desta, solicitar os vossos ofícios, no sentido de autorizar o estudante a realizar a pesquisa/recolha de dados na instituição que vossa excelência mui dignamente dirige.

Certo da vossa especial atenção de sempre, reiteramos os nossos respeitosos cumprimentos.

INSTITUTO SUPERIOR DE ANGOLA, em Luanda aos 29 de Janeiro 2024.

O Vice-Presidente Para os Assuntos Científicos

MOISÉS KAFALA NETO

Estrada da Pedreira, s/n § Bairro 17 de Setembro (Zona da Vidrui)
+244 944 588 613 § info@is-angola.com § http://www.is-angola.com
Cacucaco § Luanda § Angola §

