

Luana Filipa Manuel

**Concepção e implementação do sistema de Gestão de Pré-projectos e Monografias
Caso De Estudo – Curso de Informática da Universidade Pedagógica de Maputo**

Licenciatura em Informática

Universidade Pedagógica de Maputo

Maputo

2023

Luana Filipa Manuel

**Concepção e implementação do sistema de gestão de pré-projectos e monografias
Caso De Estudo – Curso de Informática da Universidade Pedagógica de Maputo**

Monografia a ser apresentada ao Curso de Informática, Faculdade de Engenharia e Tecnologias da Universidade Pedagógica de Maputo, para a obtenção do grau académico de Licenciatura em Informática com Minor em Eng. de Desenvolvimento de Sistemas.

Supervisora:
Mestre Cláudia Jovo Gune

Universidade Pedagógica de Maputo
Maputo
2023

Índice

Lista de Figuras.....	IV
Lista de Tabelas	V
Declaração.....	VI
DEDICATÓRIA	VIII
RESUMO	IX
Abstract.....	X
LISTA DE ABREVIATURAS	XI
CAPÍTULO I - INTRODUÇÃO.....	1
1.1. Formulação do Problema	2
1.2. Hipóteses.....	3
1.3. Justificativa	3
1.4. Objectivos	3
1.4.1. Geral.....	3
1.4.2. Específicos	3
1.5. Delimitação do universo	4
1.6. Metodologia.....	4
1.6.1. Tipo de pesquisa.....	4
1.6.2. Técnicas de recolha de requisitos.....	4
1.7. Estrutura do Trabalho	5
CAPÍTULO II – REVISÃO LITERÁRIA.....	6
2.1. Sistema.....	6
2.2. Características de sistemas.....	6
2.3.1. Sistemas fechados	6
2.3.2. Sistema Aberto	7
2.3.3. Sistema de Informação	7
2.3.3.1. Classificação dos sistemas de informação.....	8
2.4. Engenharia de Sistemas	9
2.5. Metodologia.....	11
2.5.1. Metodologias de Desenvolvimento de Sistemas.....	11
2.5.2. Tipos de Metodologia.....	11
2.5.2.1. Metodologias de Desenvolvimento estruturada	12
2.5.2.2. Metodologia orientada a objectos.....	12
2.5.2.3. Metodologia de Desenvolvimento Ágil.....	12

2.6.	Metodologia RUP – Rational Unified Process	13
2.6.1.	Características do RUP.....	14
2.6.2.	Arquitectura do RUP.....	14
2.6.3.	Metodologia OOADM	18
2.7.	Fases da Metodologia OOADM	19
2.8.	Aplicações Tradicionais vs Aplicações Web	22
2.9.	Sistema de Gestão de base de dados	24
2.10.	Características de um sistema de gestão de base de dados	25
2.11.	Linguagem de programação	26
CAPITULO III - ANÁLISE DE DISCUSSÃO DOS RESULTADOS.....		28
3.1.	Concepção e implementação do Sistema de Pré -Projectos e Monografias	28
3.1.1.	Objectivo do Sistema SGPM	28
3.1.2.	Levantamento de requisitos.....	29
3.1.3.	Identificação dos Serviços do Sistema.....	30
3.1.4.	Apresentação dos casos de Uso.....	31
3.1.5.	Representação de Diagramas	33
3.1.6.	Projecto de Interface Abstracta	36
3.2.	Implementação.....	40
3.3.	Validação dos objectivos da pesquisa.....	41
CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES.....		42
4.1.	Considerações finais	42
5.	Referências bibliográficas	43

Lista de Figuras

Figura 1: Representação do funcionamento de um SI Fonte: Laudon (2014)	8
Figura 2: ilustração das fases do RUP	14
Figura 3: Ilustração das fases da metodologia OOHDM	20
Figura 4: Apresentação da cobertura do ciclo de vida do desenvolvimento web, por cada abordagem.....	24
Figura 5: Representação do digrama de Casos de Uso	33
Figura 6: Representação do Diagrama de Classe.....	34
Figura 7: Representação do projecto navegacional (Director do curso).....	35
Figura 8: Menu Principal do SGPM	36
Figura 9: Ilustração de linhas de código do menu principal só SGPM.....	36
Figura 10: Ilustração de linhas de código do menu principal só SGPM.....	37
Figura 11: Ilustração da tabela de dados do estudante.....	37
Figura 12: Ilustração do código da tabela com dados do estudante.....	38
Figura 13: Ilustração da tabela de dados do estudante.....	38
Figura 14: Ilustração da tela de registo de estudante.	39
Figura 15: Ilustração de código de registo de estudante.....	39
Figura 16: Ilustração de código de registo de estudante	40
Figura 17: Calendário de defesas produzido pelo sistema.....	50

Lista de Tabelas

Tabela 1: apresentação de algumas metodologias de desenvolvimento	13
Tabela 2: apresentação dos requisitos funcionais do Sistema	29
Tabela 3: Apresentação dos requisitos não-funcionais do Sistema	30
Tabela 4: Descrição de casos de uso e suas dependências.....	31

Declaração

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

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

Maputo, ___08___ de ___Novembro_____ de ___2023_____

Luana Filipa Manuel

(Luana Filipa Manuel)

AGRADECIMENTOS

Em primeiro lugar agradeço ao bom Deus pelo Dom da vida e por permitir-me viver e concretizar os meus sonhos, sem a bênção de Deus nada disso seria possível.

Agradeço aos meus pais (Felicidade Júlia, Filipe Manuel, João Aleixo, Rosa Angélica e António Orlando) pelo apoio incondicional, em especial a minha querida Mãe Felicidade Júlia Maduel por ter enfrentado varias batalhas para que eu pudesse me formar.

Aos meus avós Felicidade Júlia Da Cruz e Tomas Filipe Maduel que fizeram parte da minha educação e formação, agradeço-lhes pelos melhores aprendizados e princípios que pude aprender. Aos meus tios, irmãos, primos, amigos (amigas de plantão) por todo apoio dado durante a minha carreira estudantil e por pautar que sempre fizesse as actividades estudantis atempadamente.

Agradeço a minha supervisora por ter me acompanhado e ajudado durante todo este processo, salientar que foi uma peça indispensável para a concretização desta realidade que é uma grande vitória.

Agradeço ao corpo docente da Faculdade de Engenharias e Tecnologias da Universidade Pedagógica de Maputo, que durante os 4 anos foram peças fundamentais para o conhecimento académico bem como social.

Por fim, a mim pela vontade de aprender e a determinação, agradeço aos meus colegas e amigos que estiveram comigo durante a trincheira, partilhando conhecimento, experiências, que culminaram com desenvolvimento social e profissional.

DEDICATÓRIA

Esta Monografia dedico a todos meus irmãos, Filipe Manuel Júnior, Kendall Manuel, Lira Filipe, João Kayaki, António Maduele, Sumeya Khan, Mohammed Khan, Rooney Orlando e Rhinzella Orlando que esta Monografia sirva de incentivo para cada um deles de forma a seguirem sempre os sonhos e alcançarem os seus objectivos.

RESUMO

Actualmente, as organizações públicas e privadas, procuram mecanismos sistematizados, por via de tecnologia de informação para o manuseamento e tratamento das suas actividades. As tecnologias de informação facilitam e maximizam a produtividade nos locais de trabalho, sendo que as actividades feitas manualmente podem consumir mais recursos como tempo e esforços aplicados. A Automatização visa minimizar os processos de forma a ter respostas mais rápidas e precisas.

Olhando para a UPM, especificamente para o curso de Licenciatura em Informática, sendo este um curso que visa qualificar os estudantes na área das tecnologias de informação, adequa-se a ser o local de estudo por me escolhido para aplicação dos conhecimentos adquiridos na área de formação para o enquadramento e engajamento das tecnologias de informação. A presente pesquisa teve como principal foco melhorar o processo de gestão de pré-projectos e monografias no Curso de informática na FET, com base nas tecnologias de desenvolvimento WEB, definiu-se melhor as metodologias adequadas para garantir o melhor processo na documentação da solução proposta.

Conta referir que a materialização do sistema, torna o processo eficaz e robusto em termos de resposta aos utilizadores finais, auxilia na tomada de decisões mais rápido, pois apresenta uma estrutura organizada e simplificada, também combina o monitoramento da sobreposição das datas marcadas para as defesas dos estudantes do curso de Informática na FET.

Palavra-chave: *Pré-projectos, Monografias, Sistema de gestão, tecnologias de informação.*

Abstract

Currently, public, and private organizations are looking for systematic mechanisms, through information technology, for the handling and treatment of their activities. Information technologies facilitate and maximize productivity in the workplace, and activities done manually can consume more resources such as time and effort applied. Automation aims to minimize processes to have faster and more accurate responses.

Looking at the UPM, specifically at the Degree in Informatics, which is a course that aims to qualify students in information technologies, it is suitable to be the place of study chosen by me for the application of the knowledge acquired in training for the framing and engagement of information technologies. The focus of this research was to improve the process of managing pre-projects and monographs in the Computer Course at FET, based on WEB development technologies, the appropriate methodologies were better defined to guarantee the best process in the documentation of the proposed solution.

It should be noted that the materialization of the system makes the process effective and robust in terms of responding to end users, assists in faster decision-making, as it presents an organized and simplified structure, it also combines the monitoring of the overlapping of the dates scheduled for the defenses of the students of the Informatics course at FET.

Keywords: *Pre-projects, Monographs, Management system, information technologies.*

LISTA DE ABREVIATURAS

ADV - Abstract Data View

FET - Faculdade de Engenharias e Tecnologias

GNU – General Public License

HTML – HyperText Markup Language

PHP – Personal Home Page, Hypertext Preprocessor

RAM – Random Access Memory

SGPM – Sistema de Gestão de Pré-Projectos e Monografias

SQL – Standard Query Language

CAPÍTULO I - INTRODUÇÃO

A busca de formação profissional no âmbito universitário, tem apresentado um crescimento considerável no que concerne a procura de vários cursos para a formação. Existem várias formas para a conclusão dos cursos de licenciatura. O desenvolvimento de monografia científica como forma de conclusão de curso é um processo comum para os estudantes de licenciatura.

Os estudantes em universidades de Moçambique, embora se apresentem muitas formas de culminação de curso, optam pela elaboração de monografia científica, para tal são envolvidos diversos aspectos para a sua elaboração, embora seja um trabalho que deve ser feito de uma forma individual, este envolve a participação de outros elementos que auxiliam o processo de desenvolvimento do mesmo.

Tendo em conta os aspectos que devem ser respeitados no desenvolvimento da Monografia científica, é da responsabilidade das instituições procurar mecanismos para fazer a gestão e o controle das actividades exercidas durante este processo.

No entanto, as instituições de ensino Superior enfrentam grandes desafios no que tange a gestão dos projectos e Monografias. Neste âmbito, o seguinte Projecto irá apresentar proposta de soluções para o auxílio no controle dos trabalhos de culminação de cursos, como caso de estudo escolhido a faculdade de Engenharias e Tecnologias, pertencente a Universidade Pedagógica de Maputo.

1.1. Formulação do Problema

No que concerne ao quadro de Investigação do tema em causa, observa-se que actualmente no Curso de Informática da FET (Faculdade de Engenharia e Tecnologia), que para a elaboração do trabalho de culminação de curso, os estudantes ao longo do 3º ano, escolhem dentre as varias opções apresentadas para conclusão do curso, conforme o artigo 55 do regulamento académico da Universidade Pedagógica (anexo I) que são: Exames de conclusão da Licenciatura, Monografia Científica, Pesquisa de Campo ou Compilação e Projecto, para cursos que o permitem. Actualmente a forma mais utilizada para a conclusão de curso é através de Monografia Científica, onde os estudantes seguem o modelo de elaboração de Monografia da Instituição. A fase Inicial é intitulada como a fase de Pré - projecto, que consiste na apresentação da proposta do Tema e o Supervisor que irá auxiliar no desenvolvimento da Monografia. A escolha do Supervisor tem sido feita pelos estudantes, estes em concordância, dão avanço a elaboração da Monografia Científica.

Para controlar os pré-projectos e Monografias, o responsável pelo Curso faz o registo das Monografias utilizando a plataforma EXCEL. Neste registo são apresentados dados tais como o nome do estudante, o tema proposto, respectivo supervisor e as datas de submissão e validação do pré - projecto.

Tem sido desafiador para o Curso no âmbito do controle do cumprimento dos prazos acordados, para além disso, outro aspecto relevante é sobre as estatísticas dos estudantes que estão no processo de desenvolvimento da monografia científica, bem como o número de estudantes alocados a um dado docente.

De acordo com a situação actual, nota-se que este processo é moroso no que concerne a conclusão de curso para os estudantes, o que faz com que o processo de culminação do curso muitas vezes se torne longo, diante da situação acima descrita, surge a seguinte pergunta de pesquisa:

- De que forma se pode melhorar o processo actualmente usado para a gestão dos pré - projectos e monografias?

1.2.Hipóteses

H0 (Hipótese nula) – A concepção e implementação de gestão de pré-projectos e monografias não pode melhorar o processo actualmente usado na Universidade Pedagógica de Maputo.

H1 (Hipótese alternativa) - A concepção e implementação de gestão de pré-projectos e monografias pode melhorar o processo actualmente usado na Universidade Pedagógica de Maputo.

1.3.Justificativa

Tratando-se de uma pesquisa Relacionada com a Gestão de Pré-Projectos e Monografias, justifica-se em 2 contextos:

Contexto Pessoal Concluir o curso de Licenciatura em Informática e apresentar qualidades desejadas para o desenvolvimento profissional na área de formação.

Contexto Institucional- Trazer para a Instituição um Sistema Inovador e interno no Curso de Informática da Faculdade de Engenharia de Universidade Pedagógica de Maputo, que irá permitir com que o controle dos trabalhos de culminação de cursos seja flexível, de modo a melhorar a situação actual e elevar a produtividade na Instituição Académica em prol do Desenvolvimento da Universidade Pedagógica de Maputo.

1.4.Objectivos

1.4.1. Geral

- ✓ Melhorar o processo de controle das Monografias através de um sistema de Gestão de Informação no Curso de Informática da UPM.

1.4.2. Específicos

- ✓ Integrar as Normas e políticas estabelecidas no Regulamento da Universidade Pedagógica, de modo a serem cumpridas;
- ✓ Analisar o processo actual de controlo de Monografias no Curso de Informática;

- ✓ Propor Mecanismos de segurança que possam garantir a integridade de dados no sistema;
- ✓ Implementar técnicas para a melhoria de tomada de decisão, no âmbito de alocação dos projectos.

1.5.Delimitação do universo

Tratando-se de uma unidade orgânica com múltiplas funções, o sistema fará exclusivamente gestão das Monografias na Faculdade de Engenharia. O sistema é direccionado ao Curso de Informática unidade responsável pela Gestão de Monografias do curso de Informática.

1.6.Metodologia

Para a realização do presente trabalho recorreu-se aos seguintes procedimentos metodológicos.

1.6.1. Tipo de pesquisa

Quanto à natureza, enquadra-se em uma pesquisa aplicada, pois esta voltada ao tratamento das práticas para a resolução, com aplicação de procedimentos científicos por via de revisão bibliográfica.

Quanto à abordagem, tratou-se de uma pesquisa qualitativa, visto que aprofundou-se na compreensão do processo organizacional em termos de funcionamento a nível de Gestão.

1.6.2. Técnicas de recolha de requisitos

Para a materialização do trabalho, aplicou-se uma abordagem qualitativa onde foi feita a interpretação de dados fornecidos, visto que não foram apresentados dados estatísticos, como forma de aquisição de dados utilizou-se a seguinte forma de recolha de dados:

- a) **Observação directa e participante** sobre o processo de gestão de Monografias no Curso de Informática;
- b) **Entrevista ao director do curso** de Informática da FET: que consistiu em formular uma sessão perguntas com o objectivo de obter dados que interessam a pesquisa, de

modo a perceber como tem sido o processo de gestão das Monografias e quais são os seus pontos fracos;

- c) **Questionário** dirigido ao director do Curso de Informática e aos Estudantes finalistas, que visou questionar os investigados através de uma série ordenada de perguntas que serão apresentadas no formulário.

1.7.Estrutura do Trabalho

Capitulo I: Introdução - A introdução da pesquisa, a formulação do problema de pesquisa, a justificativa, o objectivo geral e os objectivos específicos, metodologia tipo de pesquisa, Técnicas de recolha de dados.

Capitulo II: Revisão bibliográfica - Conceitos que estão directamente relacionados ao tema de pesquisa, assim como as tecnologias necessárias para a materialização do projecto.

Capitulo III: Análise e discussão dos resultados – Neste capítulo serão apresentadas possíveis soluções para a resolução do problema apresentado e detalhadamente serão apresentados os passos seguidos ate a solução.

Conclusão e Recomendações - Neste capítulo serão apresentadas as considerações finais que são a conclusão, as recomendações, trabalhos futuros e por fim, no mesmo capítulo, é apresentada a lista das referências bibliográfica que foram usadas para a materialização da pesquisa.

CAPÍTULO II – REVISÃO LITERÁRIA

Neste capítulo são apresentados os conceitos do trabalho sobre o tema em estudo com vista a familiarizar a pesquisadora com os conteúdos relacionados com o sistema de gestão de pré-projectos e monografias.

2.1.Sistema

KASPER (2000), define que um sistema é qualquer grupo de partes interagentes, inter-relacionadas ou interdependentes que formam um todo complexo e unificado que possui um objectivo específico. O principal a lembrar é que todas as partes estão inter-relacionadas e interdependentes de alguma maneira. Sem essas interdependências, temos apenas uma colecção de partes, não um sistema.

De um modo geral, Sistema é uma entidade que tem a capacidade de manter um certo grau de organização em face de mudanças internas ou externas, composto de um conjunto de elementos, em interacção, segundo determinadas leis, para atingir um objectivo específico.

2.2.Características de sistemas

Segundo KASPER (2000), as características essenciais que um sistema deve possuir são:

- Elementos;
- Relações entre Elementos;
- Objectivo Comum;
- Ambiente.

2.3. Classificação do Sistema

Os sistemas podem ser classificados de várias maneiras, porém para efeito desse trabalho, classificam-se os sistemas de duas maneiras principais: Sistemas Abertos e Sistemas Fechados.

2.3.1. Sistemas fechados

O sistema fechado é independente do meio externo para o desenvolvimento das suas funções. CORNACHIONE (1998), afirma que os sistemas fechados são entendidos como os que não mantêm relação de interdependência com o ambiente externo.

PADOVEZE (1997), afirma que os sistemas fechados não interagem com o ambiente externo, enquanto os sistemas abertos caracterizam-se pela interacção com o ambiente externo, suas entidades e variáveis.

2.3.2. Sistema Aberto

A interacção da empresa com a sociedade e o ambiente onde ela actua caracteriza essencialmente o chamado sistema aberto.

BIO (1985), propõe que os sistemas abertos envolvem a ideia que determinados *inputs* são traduzidos no sistema e, processados, geram certos *outputs*. Com efeito, a empresa vale-se de recursos materiais, humanos e tecnológicos, de cujo processamento resultam bens ou serviços a serem fornecidos ao mercado.

2.3.3. Sistema de Informação

Sistema de Informação pode ser definido como todo aquele que manipula dados e gera informação, usando ou não recursos de tecnologia de Informação.

Sistema de informação é um conjunto formado por pessoas, software, hardware, procedimentos e dados. O sistema de informação é responsável por difundir as informações através da organização (O' BRIEN, 2000).

Portanto, um SI deve ser flexível, fácil de usar, responsivo, comunicativo e rentável.

Para LAUNTON (1990) um sistema de informação pode ser definido como um conjunto de componentes relacionados trabalhando juntos para colectar, recuperar, processar armazenar e distribuir informações com a finalidade de facilitar o planeamento, o controle, a coordenação, a análise e o processo decisório em organizações.

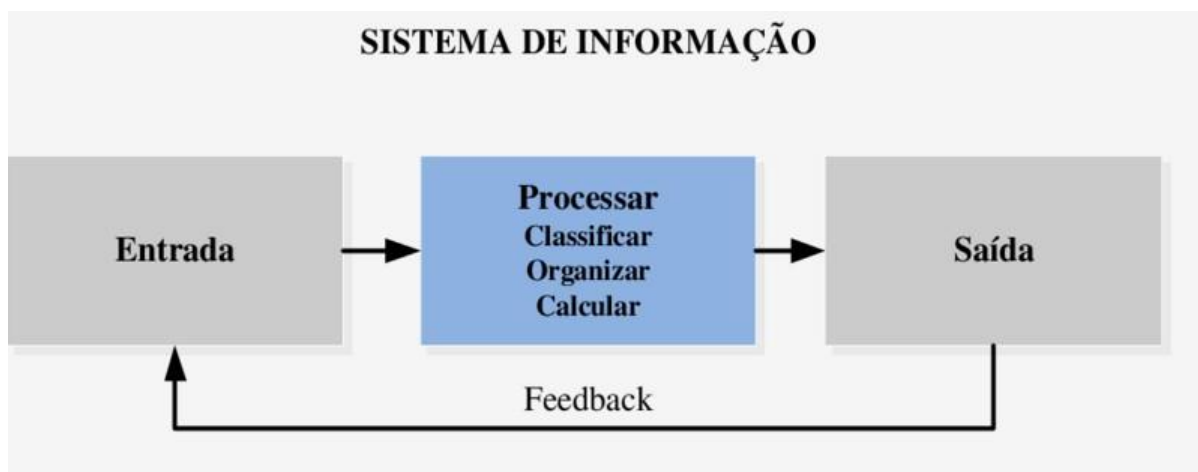


Figura 1: Representação do funcionamento de um SI
 Fonte: Laudon (2014)

Um SI pode ser visto como uma infra-estrutura que suporta o fluxo de informação interno e externo a uma organização. Ou seja interligações de componentes que trabalham de modo a colectar, recuperar, processar. Armazenar e distribuir informação com a finalidade de facilitar o planeamento, o controle, a coordenação, a análise, o processo de decisão da organização e oferece mecanismo de realimentação para atingir um objectivo.

São componentes do SI:

- ✓ Pessoas - utilizadores do sistema;
- ✓ Hardware – refere-se a componente física do SI;
- ✓ Software – parte lógica do computador cujo objectivo é processar dados;
- ✓ Redes – conjunto de dois ou mais computadores interligados com a finalidade de partilhar informação;
- ✓ Dados – matéria-prima para informação.

2.3.3.1. Classificação dos sistemas de informação

De acordo com (O' BRIEN, 2000), podemos dividir os sistemas de informação em quatro tipos:

- ✓ Sistemas de informação transaccionais;
- ✓ Sistemas de informações gerências;
- ✓ Sistemas de apoio a decisão;
- ✓ Sistemas de informações executivas.

Sistemas de informações transaccionais (SIT) - São as informações rotineiras efectuadas, como por exemplo, emissão de NF, emissão de pedido, compra de mercadoria, etc. Essas informações normalmente alimentam uma base de dados para futuras consultas.

Sistemas de informações Gerências (SIG) - Através das informações colectadas nos sistemas transaccionais, possibilitam fazer consultas e produzir relatórios para a gerência.

Sistemas de apoio à decisão (SAD) - São decisões que devem ser tomadas em um ambiente complexo, que envolve várias variáveis: localização de fornecedores, localização de clientes, impostos, política.

Sistemas de informação executiva (SIE) - os sistemas de informação executiva possuem funções dos sistemas de apoio a decisão e dos sistemas de informações gerências. Geralmente essas informações são disponibilizadas em um ambiente fácil e directo para que os executivos da empresa possam rapidamente obter uma informação.

2.4.Engenharia de Sistemas

Segundo MARTIN e McCLURE (1991) a engenharia de sistemas refere-se ao estudo dos princípios e sua aplicação no desenvolvimento e manutenção de sistemas, tanto a engenharia de sistemas como as técnicas estruturadas são colecções de metodologias de sistemas e ferramentas.

Segundo CARVALHO e CHIOSSI (2001) a engenharia de sistemas é uma disciplina que reúne metodologias, métodos e ferramentas a serem utilizados, desde a percepção do problema ate o momento que o sistema de desenvolvido deixa de ser operacional, visando resolver problemas inerentes ao processo de desenvolvimento e ao produto do sistema.

De um modo geral, ira-se aprofundar no conceito de Sommerville sobre a engenharia de sistemas.

SOMMERVILLE (1995) *define a Engenharia de software como uma disciplina de engenharia cujo foco está em todos os aspectos da produção de sistemas, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado.* Há duas expressões importantes nessa definição:

- **Disciplina de engenharia** - Engenheiros fazem as coisas funcionar. Eles aplicam teorias, métodos e ferramentas onde for apropriado. No entanto, eles os usam selectivamente e sempre tentam descobrir as soluções para os problemas, mesmo quando não há teorias e métodos aplicáveis. Os engenheiros também reconhecem que devem trabalhar de acordo com as restrições organizacionais e financeiras, então buscam soluções dentro dessas restrições.
- **Todos os aspectos da produção de** sistemas - A engenharia de sistemas não se preocupam apenas com os processos técnicos do desenvolvimento de software. Ela também inclui actividades como gerenciamento de projecto de sistemas e desenvolvimento de ferramentas, métodos e teorias para apoiar a produção de sistemas.

Engenharia tem a ver com obter resultados de qualidade requeridos dentro do cronograma e do orçamento.

Em geral, os engenheiros de sistemas adoptam uma abordagem sistemática e organizada para seu trabalho, pois essa costuma ser a maneira mais eficiente de produzir sistemas de alta qualidade. No entanto, engenharia tem tudo a ver com seleccionar o método mais adequado para um conjunto de circunstâncias, então uma abordagem mais criativa e menos formal pode ser eficiente em algumas circunstâncias. Desenvolvimento menos formal é particularmente adequado para o desenvolvimento de sistemas Web, que requerem uma mistura de habilidades de sistemas e de projecto.

A Engenharia de Sistemas tem como elementos:

- **Método** – procedimento formal para a produção de um resultado. Um método define as tarefas que serão realizadas e de que forma as mesmas deverão ser executadas. Define que tecnologias e linguagens deverão ser adoptadas e os produtos que serão produzidos.
- **Ferramenta** – instrumento ou sistema automatizado (software) para suportar a realização de uma tarefa.
- **Paradigma** – uma maneira, estilo ou filosofia para a construção de sistemas, envolvendo um conjunto de princípios.

2.5. Metodologia

“A metodologia é a lógica dos procedimentos científicos em sua gênese e em seu desenvolvimento, não se reduz, portanto, a uma “metrologia” ou tecnologia da medida dos fatos científicos” (LAKATOS e MARCONI, 2003)

2.5.1. Metodologias de Desenvolvimento de Sistemas

Diversas metodologias foram criadas para sistematizar o desenvolvimento de softwares. Essas metodologias podem ser divididas em *tradicionais*, que enfatizam a documentação de cada passo do desenvolvimento do software, ou *ágeis*, consideradas um paradigma novo de desenvolvimento de sistemas.

Uma metodologia de desenvolvimento de sistemas é um conjunto de actividades que auxiliam a produção de sistemas. O resultado dessas actividades é um produto que reflecte a forma como todo processo foi conduzido. Embora tenham sido criadas varias metodologias para o desenvolvimento de sistemas, existem actividades fundamentais comuns a todas elas (SOMMERVILLE, 2007):

- **Especificação:** definição das funcionalidades e demais características do produto;
- **Projecto e implementação:** o sistema é produzido de acordo com as especificações. Nesta fase são propostos modelos por meio de diagramas que serão implementados em alguma linguagem de programação;
- **Validação:** actividade de revisão e testes visando a assegurar que os requisitos sejam cumpridos;
- **Evolução:** actividades de manutenção por exemplo para adaptar o sistema a nova necessidade do cliente.

2.5.2. Tipos de Metodologia

SOMMERVILLE (1995), considera que a maior parte do desenvolvimento de sistemas é uma actividade profissional, em que o sistema é desenvolvido para um propósito específico de negócio, sistema de informação, sistema CAD, entre vários outros. Para tal, considera-se também que o desenvolvimento do sistema apresenta exigências profissionais para a sua elaboração e implementação, para que o sistema computacional vise transformar as necessidades do utilizador em um sistema informatizado, através do uso das tecnologias de

informação. As metodologias de desenvolvimento de sistemas podem ser classificadas em estruturada, orientadas a objectos e ágil.

2.5.2.1. Metodologias de Desenvolvimento estruturada

As metodologias estruturadas são também chamadas de orientadas a documentação. No âmbito do surgimento tinha como objectivo formalizar o processo de requisitos, de modo a reduzir as possibilidades de má interpretação dos mesmos e introduzir técnicas baseadas nas melhores praticas ao processo de análise e desenho. A metodologia estruturada é uma aplicação de conjuntos de princípios usados que tendem a construção de modelos que retratam o fluxo de informações.

2.5.2.2. Metodologia orientada a objectos

A metodologia mostra uma colecção de classes e objectos relacionados. A perspectiva de modelação dos sistemas muda, uma vez que o mesmo conceito base é utilizado ao longo de todas as fases do processo, promovendo a reutilização e o encapsulamento da informação, e facilitando a manutenção.

2.5.2.3. Metodologia de Desenvolvimento Ágil

Os métodos ágeis caracterizam-se pelo seu carácter adaptativo e orientado para pessoas.

Segundo SEMEDO (2012), os Processos Ágeis de Desenvolvimento compartilham a premissa de que o cliente aprende sobre suas necessidades, na medida em que é capaz de manipular o sistema que está sendo produzido e, com base no feedback do sistema, ele reavalia as suas necessidades e prioridades, gerando mudanças que devem ser incorporadas ao software. O aprendizado é importante, porque permite que o cliente direcione o desenvolvimento de modo que a equipe produza sempre aquilo que tem o maior valor para o seu negócio.

As abordagens ágeis compartilham, na sua essência, o processo de desenvolvimento centrado nas pessoas, orientado para a obtenção de artefactos a partir de iterações, o que, consequentemente, impõe o carácter adaptativo durante todo o ciclo de desenvolvimento.

Existem várias metodologias, que podem ser implementadas, destacam-se as seguintes:

Tabela 1: apresentação de algumas metodologias de desenvolvimento

Metodologia Estruturada	Metodologia Orientada a Objectos	Metodologia Ágil	Outras Metodologias
SADT	OOHDM - Object-Oriented Hypermedia Design Method;	Feature Driven Development - FDD	Microsoft Solution Framework - MSF
Diagrama de fluxo de dados-DFD	Rational Unified Process - RUP	Enterprise Unified Process –EUP	
Programação estruturada	UWE – UML-Based Web Engineering	Scrum	
Modelo de Entidade e Relacionamento – MER	XP – Extreme Programming	Crystal - (Crystal Clear, Crystal Orange, Crystal Orange Web)	
Análise Estruturada	WebML		
Projecto estruturado	OPEN – Object-oriented Process, Environment and Notation		
Análise essencial			

Fonte: SEMEDO (2012)

As metodologias apresentadas na tabela, podem ser implementadas segundo as necessidades.

2.6. Metodologia RUP – Rational Unified Process

RUP é resultado do desenvolvimento e refinamentos realizados pela Rational Corporation sobre Processo unificado (PU), (PRESSMAN, 2006).

O RUP baseia-se no paradigma de Orientação a Objectos, projecto e documentado utilizado a notação Unified Modeling Language (UML), para a ilustração dos processos. É considerado como um conjunto de práticas colectadas de engenharia de software que são continuamente aprimoradas, com regularidade, para reflectirem alterações nas práticas do segmento do mercado, fornece técnicas de desenvolvimento de software de modo a aumentar a sua produtividade.

2.6.1. Características do RUP

O RUP suporta diversas boas práticas do desenvolvimento de software das quais, destacam-se as seguintes:

- ✓ **Guiado por casos de Uso**- processo de desenvolvimento segue um fluxo, em que os casos de utilização são especificados, desenhados, implementados e no fim são a fonte a partir dos quais os testes são definidos e realizados;
- ✓ **Baseado na arquitectura do sistema** – permite implementar os casos de uso requisitados;
- ✓ **Metodologia interactiva e incremental** – projecto facilmente gerido e executado se for dividido em várias partes ou mini projectos, em que cada mini projecto é uma iteração que resulta num incremento, que devera ser devidamente planeado, controlado e executado.

2.6.2. Arquitectura do RUP

O RUP apresenta dois eixos, como mostra a figura a baixo, duas visões através das quais um sistema é descrito.

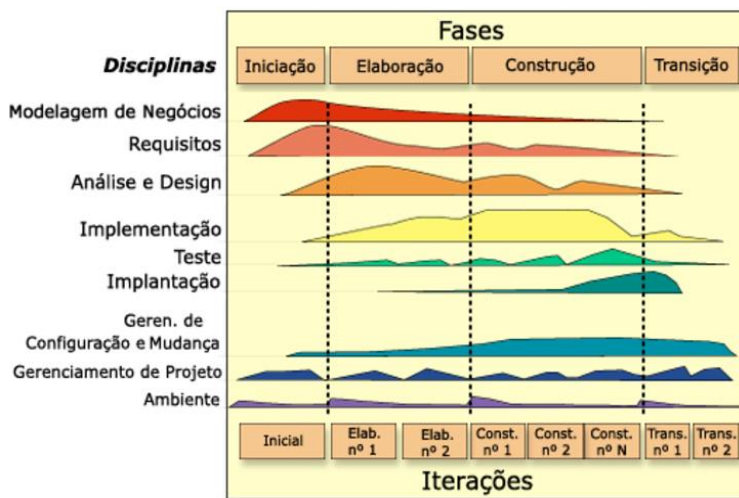


Figura 2: ilustração das fases do RUP
Fonte: Adaptado por, SILVA & VIDEIRA (2011)

1. Concepção

Fase inicial do processo de desenvolvimento do software, é também uma fase de grande importância, principalmente para os novos projectos, os quais existe um grande número de requisitos que necessitam de uma especificação clara. Esta fase dá ênfase ao tratamento de riscos relacionados com caso de negócio específico, onde a tarefa de qualificação de riscos de um projecto passa a se tornar prioridade essencial, o âmbito do projecto e são identificadas todas as entidades externas com as quais o sistema interage.

➤ **Requisitos**

Segundo SOMMERVILLE (2007), requisitos são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições ao seu funcionamento.

Os requisitos de sistema podem representar uma necessidade funcional do sistema, ou uma característica não funcional essencial ao software, como um nível de segurança. Os requisitos do sistema são classificados em funcionais e não funcionais:

a) Requisitos Funcionais

“Os requisitos funcionais de um sistema descrevem o que o sistema deve fazer.”
(SOMMERVILLE, 2007)

Os requisitos funcionais de um sistema reflectem as funcionalidades principais do sistema especificando, de forma clara, o que o sistema deve fazer em cada acção feita sobre o sistema. É crucial que a especificação dos requisitos cubra todos os serviços exigidos pelo utilizador final.

b) Requisitos não Funcionais

Os requisitos não funcionais detalham como os requisitos funcionais serão implementados. Estes requisitos estão associados às propriedades emergentes do sistema como a fiabilidade, segurança, tempo de resposta, espaço de armazenamento, robustez. Normalmente são solicitados pelos gestores de projectos e estão relacionados com restrições sobre os quais os utilizadores devem estar satisfeitos.

Os principais resultados desta fase são:

- Documento com a visão geral do problema: requisitos principais, funcionalidades mais significativas, restrições importantes;
- Modelo de casos de utilização que são especificados;
- Glossário inicial do projecto;
- Relatório inicial do projecto;
- Justificação de negócio (business case) inicial;
- Plano de projecto;
- Protótipos.

Na fase de concepção são usadas ferramentas que auxiliam na materialização dos requisitos e objectivos traçados. Para tal recorre-se a ferramentas para modelação gráfica do sistema, fornecidas pela linguagem UML.

2. Elaboração

Nesta fase o foco está virado aos riscos técnicos e de arquitectura, em que o escopo do projecto deve ser revisto e os requisitos devem estar claros. Busca complementar a documentação dos casos de uso.

O propósito desta fase é analisar o domínio do problema, desenvolver o plano de projecto, estabelecer a fundação arquitectural e eliminar os elementos de alto risco. Os elementos de risco a serem analisados, nesta fase, são os riscos de requerimentos, tecnológicos (referentes a capacidade das ferramentas disponíveis), de habilidades (dos integrantes do projecto) e políticos. Esta é a fase mais crítica de todas, pois ao final desta fase a engenharia é considerada completa e os custos para modificação do sistema aumentam a medida que o projecto avança. Do ponto de vista administrativo, é ao final desta fase que um projecto deixa de ser uma operação de baixo risco e baixo custo para se tornar uma operação de alto risco e alto custo.

3. Construção

Esta fase dá ênfase ao desenvolvimento físico do software. Pode-se dizer que é a fase de transição da construção do produto, entre a concepção e a elaboração para a distribuição. Nesta fase o resultado esperado é um produto pronto para os utilizadores finais, com um produto de software integrado nas plataformas adequadas, manual de usuários e uma descrição da versão actual.

4. Transição

A fase de implementação, ocorre a entrega do software e é realizada a capacitação dos utilizadores finais do produto desenvolvido.

Nesta fase é feito um refinamento do projecto com base no feedback dos utilizadores, como configuração, instalação e problemas de usabilidade. Outros factores a serem considerados são requisitos adicionais, características que podem vir a surgir no final do ciclo de vida do projecto, o que significa que, haverá novas iterações, exigindo uma nova análise, design etc., semelhante a fase de construção.

A fase de transição pode ser a entrega e o término de um projecto, também pode significar o início de um novo ciclo de vida do mesmo projecto, o que poderia ser considerado como uma nova versão do produto.

I. Componente Estática: Workflows

A visão estática do RUP descreve as tarefas (Workflows) e as actividades realizadas, bem como os artefactos produzidos e os respectivos intervenientes. Esta visão encontra-se centrada no conceito de workflow. No RUP os Workflows encontram-se divididos em Workflows de processamento e Workflows de suporte. No primeiro grupo incluem-se os Workflows:

a) Modelagem de negócios (Business Modeling)

A modelagem de negócios trata de descrever como desenvolver uma visão da nova organização, definindo processos, papéis e responsabilidades de cada um dos envolvidos através de um caso de uso em um modelo de objectos de negócio. A visão do negócio identifica o que poderá ser alterado de acordo com a evolução do entendimento e objectivos a serem alcançados.

b) Requisitos (Requiements)

A disciplina de requisitos procura trabalhar de forma a identificar as principais necessidades dos clientes para que possam ser bem compreendidas por todos, negociadas, aceites e gerenciadas durante o desenvolvimento do sistema, visando reduzir os riscos para o desenvolvimento do sistema através de métodos e processos para identificação e documentação dos requisitos do sistema.

c) Análise e Desenho (analysis and Design)

Envolve actividades de definição do modo como o sistema será construído na fase de implementação. Esta disciplina tem como objectivos:

- Transformar os requisitos em um design do sistema a ser criado;
- Desenvolver uma arquitectura sofisticada para o sistema;
- Adaptar o design para que corresponda ao ambiente de implementação, projectando para fins de desempenho.

d) Implementação (implementation)

Através deste workflow pretende-se definir a estrutura do sistema e implementar os respectivos componentes; testar individualmente cada um; e efectuar a integração dos vários componentes, de modo a construir um sistema executável. São por isso realizadas tarefas

associadas a programação e integração, o que torna necessária a participação de intervenientes do tipo programador e integrador de sistemas, com o apoio do arquitecto e do revisor de código. Os principais artefactos a produzir por este workflow são os seguintes:

- Componente, que são unidades de código, elementares ou que resultam da agregação de outros componentes;
- Subsistemas de implementação, construídos por uma colecção de componentes e outros subsistemas de implementação. O seu conjunto constitui o sistema global;
- Plano de integração de uma release parcial, que define a ordem de desenvolvimento e integração de componentes.

e) **Testes (Test)**

Actividades de verificação de todo o sistema, garante:

- Integração entre os componentes implementados;
- Correção dos erros detectados antes da instalação do sistema;
- Conformidade entre o sistema implementado e os requisitos definidos pelos utilizadores.

f) **Instalação (Deploy)**

De forma a garantir o sucesso do sistema desenvolvido, é essencial efectuar a sua disponibilização para os seus utilizadores finais da forma mais adequada e com o mínimo impacto no seu trabalho. É este o objectivo do workflow de instalação. De modo a que o objectivo seja alcançado, esta prevista a realização de diversas actividades de apoio, como sejam a produção de manuais de formação e de utilização do sistema. ~

2.6.3. Metodologia OOHD

A metodologia OOHD é uma abordagem de desenvolvimento de aplicações hipermédia baseada em modelos. O método compreende 4 etapas, onde o modelo da aplicação é construído e enriquecido a cada passo. Cada etapa é baseada na anterior, cujo estilo de desenvolvimento é interactivo e incremental, (SCH 94b).

O modelo de autoria OOHD apresenta uma abordagem orientada à objectos, separando a modelagem estrutural da implementação, propiciando o desenvolvimento de aplicações hipermédia, como web, sistemas de informação, ambientes virtuais de aprendizagem, simuladores educacionais, sites, entre outros. (SCHWABE & ROSSI,1999).

Conforme CUNHA (2002), o método OOHDm permiti o desenvolvimento gradativo de uma aplicação hipermédia através de níveis distintos de abstracção até uma descrição em alto nível. Esse método é um mecanismo de modelagem que compreende quatro etapas, onde a aplicação é desenvolvida e melhorada em cada uma delas, permitindo a descrição do domínio da aplicação utilizando mecanismos de alto nível.

A modelagem realizada é orientada a objectos, proporcionando um alto nível de reuso e manutenção, e favorecendo a legibilidade e comunicação entre os desenvolvedores envolvidos. O mecanismo de modelagem OOHDm permite que um domínio da aplicação seja descrito através de mecanismos de alto nível independente do sistema de hipertextos já desenvolvido ou a ser projectado.

Este modelo possibilita que a elaboração do projecto conceitual não se preocupe demasiadamente com os detalhes de implementação. O OOHDm independe da definição da linguagem e ambiente de programação, permitindo que a elaboração do projecto seja mais ampla e concreta.

LIMA (2003) diferencia o OOHDm dos outros métodos tradicionais da engenharia de software para o desenvolvimento de aplicações hipermédia através do alto grau de importância e significância dado ao aspecto navegacional presente no modelo, a fim de possibilitar a realização de melhorias e mudanças em cada etapa de um modelo para o outro.

2.7. Fases da Metodologia OOHDm

O modelo OOHDm contempla 5 fases, que são respectivamente:

- ✓ Levantamento de requisitos;
- ✓ Modelagem conceitual;
- ✓ Projecto navegacional;
- ✓ Projecto de interface abstracta;
- ✓ Implementação.

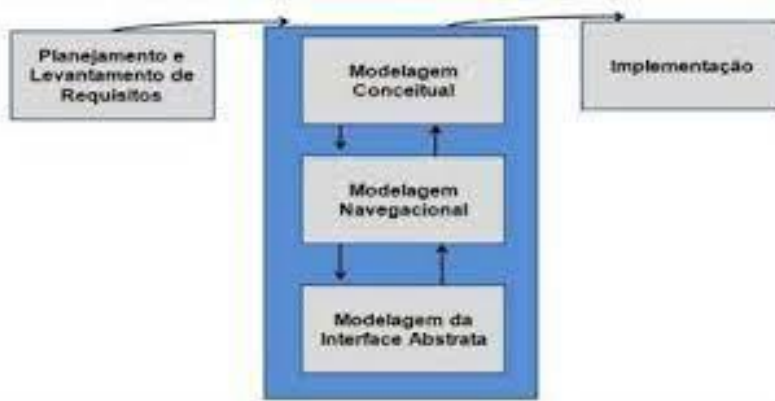


Figura 3: Ilustração das fases da metodologia OOHDM

Fonte: SCHWABE E ROSSI, 1994

a) Levantamento de Requisitos

A etapa de levantamento de requisitos consiste na identificação dos usuários da aplicação que será desenvolvida e na definição das tarefas que serão apoiadas pelo aplicativo. Segundo SCHWABE et al (1994), esta etapa envolve as seguintes fases: identificação dos atores e tarefas, especificação dos cenários, especificação dos casos de uso, especificação dos UIDs (diagrama de interação dos usuários), e validação dos casos de uso e dos UIDs.

Esta etapa consiste em obter o máximo de informações possíveis referentes sobre o domínio da aplicação. Informações como matérias a serem disponibilizados, temas a serem trabalhados, recursos disponíveis, níveis de interação do usuário e diferença de tarefas para cada usuário devem ser obtidas.

b) Modelagem conceitual

Segundo FRANÇA (2005), na etapa de modelagem conceitual é realizada a abstracção do problema para a elaboração de uma base conceitual, a fim de representar objectos e relacionamentos existentes no domínio da aplicação, sem levar em consideração ainda os aspectos navegacionais ou relativos à plataforma de implementação.

Nesta etapa é elaborado um esquema conceitual constituído sobre classes, relações, atributos e subsistemas, seguindo os princípios dos modelos orientados a objectos. Assim, é possível representar os modelos desenvolvidos através da notação UML (Unified Modeling Language), porém, os seus atributos podem ser multi-tipados, ou seja, podem representar várias perspectivas sobre determinada entidade real do domínio.

c) Projecto navegacional

Uma aplicação hipermédia disponibiliza um espaço de informações para que o usuário navegue através dele. Com isso, o projecto navegacional da aplicação é a etapa crucial no empreendimento de desenvolvimento (SCHWABE E ROSSI, 2002). Nesta etapa é significativamente importante destacar como o usuário pode explorar o ambiente criado na aplicação de forma a evitar a redundância de informações e orientar o usuário na utilização da aplicação.

De acordo com PEREIRA (2006), nesta fase do OOHDM o projectista do aplicativo hipermédia deve pesquisar e conhecer o perfil do usuário e as tarefas que o mesmo poderá executar na aplicação proposta. Ocorre nesta fase um mapeamento completo do modelo conceitual em objectos navegacionais. Esse mapeamento resulta em um conjunto de nós, elos, estruturas de acesso e contextos navegacionais da aplicação hipermédia.

Os nós são definidos como as informações básicas, que possuem atributos e âncoras. As ligações definem os relacionamentos a serem utilizados pelos usuários. As estruturas de acesso funcionam como índices ou dicionários de dados e são úteis na procura pela informação desejada pelos usuários (PEREIRA, 2006).

Segundo SILVA (2012), nesta fase de projecto de navegação, do modelo OOHDM, é realizado uma descrição completa das estruturas de acesso do aplicativo hipermédia. A descrição é feita de forma conceitual, ou seja, de acordo com o nível de abstracção do domínio da aplicação hipermédia proposta. Ocorre nesta fase também, o estabelecimento de padrões gerais de navegação para a descrição da estrutura geral da aplicação hipermédia a ser desenvolvida.

d) Projecto de Interface Abstracta

Projecto de Interface Abstracta Segundo BABA (2005), o projecto de interface abstracta descreve os objectos de interface que podem ser vistos, suas propriedades e transformações durante a navegação, de forma a descrever as interacções do usuário com a aplicação. Nesta etapa envolve a abordagem de projecto Abstract Data View (ADV) para descrever a interface de uma aplicação hipermédia.

De acordo com BRITO (2003), ADVs são modelos formais que seguem o padrão da orientação a objectos, com objectos de interface (menus, campos de texto, botões etc.) que

possibilitam a especificação das metáforas de interface e a descrição das propriedades estáticas (relacionamentos com os objectos 37 navegacionais) e dinâmicas (comportamento frente a eventos externos), independente de implementação.

O desenvolvimento de uma interface hipermédia é um aspecto extremamente crítico e crucial na criação de qualquer aplicação hipermédia. Para especificar um modelo de abstracto de interface é necessário definir seus aspectos, suas propriedades estáticas e dinâmicas e seus relacionamentos com o modelo navegacional proposto, independente da etapa de implementação.

e) Implementação

Implementação Segundo PEREIRA (2006), a implementação é a actividade final proposta pelo método OOADM, responsável pela tradução directa dos modelos de aplicação especificados para um ambiente de utilização. A tradução é realizada em uma plataforma de hardware e software escolhida pelo desenvolvedor da aplicação de forma a resultar em um produto final, que será utilizado para a execução plena do aplicativo hipermédia proposto. A implementação mapeia o modelo conceitual, de interface abstracta e navegacional para objectos concretos disponíveis no ambiente de desenvolvimento escolhido, de forma a obter uma aplicação final para ser executada.

Segundo CUNHA (2002), “a implementação de uma aplicação hipermédia não é uma tarefa simples, como resolver as questões técnicas e não técnicas da aplicação”.

Conforme ZAMBALDE & ALVES (1999), a implementação é a etapa final do OOADM, responsável por traduzir o projecto da aplicação para um ambiente específico de desenvolvimento, a fim de executar a aplicação em uma plataforma de hardware e software escolhida.

2.8. Aplicações Tradicionais vs Aplicações Web

Uma aplicação Web é um sistema que usa internet como meio de comunicação entre o cliente e o sistema. Define-se uma aplicação Web como uma aplicação de software que utiliza a Web como ambiente de execução. Aplicações Web envolvem Sites Web ou sistemas Web (DE ABREU, 2016).

Uma aplicação Web deve ser entendida como uma forma de uso de software acessando dados persistentes através do serviço Web, permitindo a construção dinâmica de páginas para manipular estes dados.

As principais diferenças entre aplicações Web e aplicações tradicionais referem-se a questões de navegação, organização da interface e implementação, (ROSSI, 1999).

Porém, alguns autores defende que não existe uma metodologia que cobre todo ciclo de vida do desenvolvimento Web.

A maioria das metodologias da Web actualmente existentes concentram-se mais nos aspectos de design e não na engenharia de requisitos, embora os riscos de uma definição e validação incompleta ou insuficiente de requisitos sejam bem conhecidos.

A metodologia RUP é considerada uma metodologia robusta que foi desenvolvida com o foco nas aplicações tradicionais, porem estuda na sua integra os detalhes para desenvolvimento de um sistema, aspectos como interface, navegabilidade não são mencionados na metodologia RUP, portanto há necessidade de usa-las de forma adaptada com o auxílio de outras metodologias que focam em detalhes necessários para o desenvolvimento de aplicações Web, que possam cobrir o ciclo de vida do sistema.

FERREIRA (2013), numa das suas abordagens, apresenta uma proposta para modelagem da navegação e da apresentação de aplicações Web, uma extensão UML que e parte de uma metodologia para análise e projecto de aplicações web baseada no *OOHDM – Object Oriented Hypermedia Design Method*, esta metodologia trata separadamente o conteúdo, a navegação e a apresentação da aplicação, através da execução dos seguintes passos: Projecto conceitual, Projecto navegacional e Projecto da Apresentação.

Sistemas de Gestão de bases de dados são programas que permitem criar e manipular bases de dados, em que os dados são estruturados com independência relativamente aos programas de aplicação que os manipulam. Os SGBD apresentam 3 níveis a respeitar na sua arquitectura que são:

- ✓ **Nível físico:** é constituído pelo armazenamento dos ficheiros de dados em suportes informáticos e a forma como eles se encontram organizados nesse suporte;
- ✓ **Nível conceptual:** organização da informação em entidades e relacionamentos;
- ✓ **Nível de visualização:** ecrãs com interfaces que apresentam a informação da base de dados.

Existem vários SGBD dos quais como exemplo destes os seguintes:

- ✓ IBD Informix;
- ✓ PostgreSQL;
- ✓ HSQLBD;
- ✓ MySQL;
- ✓ SQL-Server;
- ✓ Microsoft Access; e
- ✓ Outros.

2.10. Características de um sistema de gestão de base de dados

- ✓ Controlar a redundância de dados;
- ✓ Compartilhando de dados;
- ✓ Segurança;
- ✓ Backup e recuperação a falhas;
- ✓ Forçar restrições de integridade;
- ✓ Aumentar a produtividade e disponibilidade;
- ✓ Flexibilidade, padronização;
- ✓ Independência de dados.

MySQL

“O MySQL é um SGBD que usa a linguagem SQL como interface, é de código fonte aberto com uma comunidade de desenvolvedores a nível mundial e também protegido por uma licença GNU.” (SUEHRING, 2002).

É um sistema de gestão de base de dados relacionais de código fonte aberto geralmente utilizado em aplicações Web devido à velocidade, flexibilidade e fiabilidade, emprega a linguagem SQL para processar dados contidos na base de dados.

2.11. Linguagem de programação

Uma linguagem de programação é um método padronizado que usa-se para expressar as instruções de um programa a um computador programável. Ela segue um conjunto de regras sintáticas e semânticas para definir um programa de computador. Regras sintáticas dizem respeito à forma de escrita e regras semânticas ao conteúdo, (SEBASTA, 2000).

Através da especificação de uma linguagem de programação pode-se especificar quais dados um computador vai usar; como estes dados serão tratados, armazenados, transmitidos; quais acções devem ser tomadas em determinadas circunstâncias.

Para o processo de desenvolvimento de um sistema, os programadores tem a função de escolher que linguagem de programação pretende usar para a construção do sistema. Existem diversas linguagens de programação, dentre elas temos nomeadamente:

- ✓ C;
- ✓ Java;
- ✓ JavaScript;
- ✓ PHP;
- ✓ Visual Basic;
- ✓ C#.

Para a concepção do sistema requerido, tratando-se de um sistema Web, usar-se-á o PHP, que é a base para a construção de sistemas Web e envolve dinamismo. Associada a esta tecnologia ira-se implementar tecnologias como o JavaScript e o CSS para complementar o processo de desenvolvimento Web.

a) PHP

PHP é uma linguagem que permite criar Sites WEB dinâmicos, possibilitando uma interacção com o usuário através de formulários parâmetros da URL e hiperligações. A diferença de PHP com relação a linguagens semelhantes a JavaScript é que o código PHP é executado no servidor, sendo enviado para o utilizador apenas HTML puro.

b) JavaScript

O JavaScript é importante, pois fornece funcionalidade dinâmica no navegador e, através do Ajax, comunicação oculta com o servidor Web “modificar dados da interface sem recarregamento das páginas”. (NIXON, 2012).

c) CSS

O Cascading Style Sheets (CSS) é uma folha de estilo composta por camadas e utilizada para definir a apresentação em páginas Web que adoptam para o seu desenvolvimento linguagens de marcação.

O CSS define como serão exibidos os elementos contidos no código de uma página da internet e sua maior vantagem é efectuar a separação entre o formato e o conteúdo de um documento.

Unified Modeling Language (UML)

“O UML (Unified Modeling Language) é uma linguagem para especificação, construção, visualização e documentação de artefactos de um sistema de software.” (SILVA & VIDEIRA, 2001)

Os objectivos da UML são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.

Segundo RAMOS (2011), A UML fornece vários diagramas que auxiliam no desenvolvimento dos modelos gráficos, nomeadamente:

Diagramas estruturais (estáticos)

- Diagramas de classe e diagrama de Objectos;
- Diagramas de componentes;
- Diagramas de instalação;
- Diagramas de pacotes;
- Diagramas de implementação;

Diagramas comportamentais (dinâmicos)

- Diagramas de actividades;
- Diagramas de caso de uso;
- Diagramas de estados;
- Diagramas de interacção (diagramas de sequencia e diagramas de colaboração).

CAPITULO III - ANÁLISE DE DISCUSSÃO DOS RESULTADOS

3. Descrição do Local de Estudo

A FET é uma unidade orgânica técnica e académica que visa formar indivíduos capacitados para a formação nos níveis do ensino técnico, profissionais superiores assim como de outras áreas técnicas. A FET localiza-se em Maputo, Campus Universitário Lhanguene Avenida de Moçambique, Kml.

Esta Unidade orgânica foi criada em Fevereiro de 2008, e tem como missão a construção, transmissão e divulgação de conhecimentos com o intuito de promover a formação de nível superior nos campos de engenharia, das artes, da agro-pecuária e das ciências tecnológicas, num relacionamento dinâmico e amplo com a indústria e o mundo do trabalho. A FET provedora de programas de engenharia, tecnologia e artes gráficas com ancoragem no ensino, pesquisa e actividades de intervenção comunitária (extensão).

a) Curso de Informática

O Curso de Informática (CI) é um sector subordinado pela FET e tal como missão o ensino, a investigação e a transferência de conhecimento e inovação na área das Ciências de Computação.

O Curso de Informática ministra formação de nível superior da graduação e pós-graduação, organizando cursos dos graus académicos correspondentes na área da informática, tendo como destaque a Engenharia de Programação e Comunicação e Redes de Computadores.

3.1. Concepção e implementação do Sistema de Pré -Projectos e Monografias

3.1.1. Objectivo do Sistema SGPM

O Sistema tem com objectivo a dinamização do processo de Gestão de pré - projectos e monografias científicas, visto que a gestão dos mesmos está associada ao tratamento e armazenamento de informação, este foi desenvolvido olhando para as regras de negócios e os requisitos, de forma a auxiliar os utilizadores finais na tomada de decisão de forma pratica e rápida.

3.1.2. Levantamento de requisitos

Considerando aquilo que é uma das metodologias escolhidas RUP, esta que pressupõe as seguintes fases: Concepção, elaboração, construção e transição, teremos em conta a primeira fase de Concepção para a apresentação dos requisitos do sistema, que estão divididos em dois tipos, requisitos funcionais e requisitos não funcionais.

a) Requisitos funcionais

Considera-se como requisitos funcionais, aqueles que influenciam directamente no funcionamento do sistema:

- Os serviços que o sistema deve fornecer;
- Como sistema deve reagir uma entrada ou acção pretendida pelo utilizador;
- O comportamento do sistema a determinadas situações.

Tabela 2: apresentação dos requisitos funcionais do Sistema

Requisitos	Descrição	Operação
RF01	Gerir utilizadores	Registar, consultar e editar dados dos utilizadores
RF02	Gerir Curso	Registar, consultar e editar dados dos cursos
RF03	Gerir Minor	Registar, consultar e editar dados do Minor
RF04	Gerir Estudante	Registar, consultar e editar dados do estudante
RF05	Gerir Docente	Registar, consultar e editar dados do Docente
RF06	Gerir Pré-Projecto	Registar, consultar e editar dados do Pré – Projecto
RF07	Gerir Monografia	Registar, consultar, editar dados da Monografia e solicitar aprovação
RF08	Marcação da Defesa	Marcar data, escolher os membros do júri e o lugar da defesa

Fonte: Autor

b) Requisitos não funcionais

Tabela 3: Apresentação dos requisitos não-funcionais do Sistema

Requisito	Descrição
Segurança	O acesso aos dados do sistema só será disponibilizada após a autenticação do Utilizador e consoante ao seu nível de acesso, como esta prescrito na regra de negocio(RN).
Disponibilidade	O sistema compete a utilização de Web Browser(Internet Explorer, Mozilla firefox ou Google Chrome) e acesso a Internet.
Capacidade	O sistema deve suportar mais de 5 utilizadores simultaneamente, sem nenhuma falha de sistema.
Integridade	O sistema deve garantir a autenticidade e veracidade da informação por ele gerada.

Fonte: Autor

3.1.3. Identificação dos Serviços do Sistema

Tendo em conta a apresentação dos requisitos do sistema é possível identificar os serviços que serão disponibilizados pelo sistema:

- ✓ Registos de projectos de culminação do curso e parametrização de datas para a validação;
- ✓ Validação dos projectos para a fase de monografia;
- ✓ Registo das monografias e parametrização de datas para validação;
- ✓ Monitoria das datas parametrizadas;
- ✓ Validação das monografias para a fase de defesa/apresentação final;
- ✓ Calendário das defesas;
- ✓ Datas estatísticos tais como: número de estudantes e projectos registados, número de defesas marcadas.

3.1.4. Apresentação dos casos de Uso

Abaixo serão apresentados os casos de uso que foram compilados a partir dos requisitos do sistema. Nestes serão identificadas as dependências dos casos de usos com outros caso de Uso.

Tabela 4: Descrição de casos de uso e suas dependências

Requisito	Caso de Uso Mãe	Descrição	Casos de Uso filhos	Dependência
RF01	CDU01	Gerir utilizadores	CDU02–Registrar Utilizadores; CDU03 – Visualizar dados do Utilizador; CDU04 – Editar dados do Utilizador	CDU03 & CDU04 dependem de:CDU02
RF02	CDU05	Gerir Curso	CDU06 – Registrar curso CDU07 – visualizar CDU08 - Editar curso.	CDU07 & CDU08 dependem de :CDU06
RF03	CDU09	Gerir Minor	CDU10 – Registrar minor CDU11 – Visualizar minor CDU12 – Editar minor	CDU09 depende de CDU05; CDU11 & CDU12 Dependem de: CDU10
RF04	CDU13	Gerir Docente	CDU14 – Registrar docente CDU15 – Visualizar dados do docente CDU16 – Editar dados de docente	CDU15 & CDU16 Dependem de: CDU14
RF05	CDU17	Gerir Estudante	CDU18 – Registrar estudante CDU19 – Visualizar dados do estudante CDU20 – Editar dados de estudante	CDU19 & CDU20 dependem de: CDU18
RF06	CDU21	Gerir Projecto	CDU22 – registar projecto CDU23 – visualizar dados do projecto CDU24 – Editar dados do projecto	CDU23 & CDU24 dependem de: CDU22
RF07	CDU25	Gerir Monografia	CDU26 – registar monografia CDU27 – Visualizar dados da monografia CDU28 – Editar dados da Monografia	CDU25 depende de CDU21; CDU27 & CDU28 dependem de: CDU26
RF08	CDU29	Gerir Defesa	CDU30 – Registrar Defesa CDU31 – Visualizar dados da defesa CDU32 – Editar dados da defesa	CDU29 depende de CDU25 CDU31 & CDU32 dependem de: CDU30

Fonte: Autor

3.1.5. Representação de Diagramas

Nesta fase serão apresentados os diagramas do sistema, o diagrama de Caso de Uso, o diagrama de Classe e o Diagrama de Navegação.

A) Diagrama de caso de uso

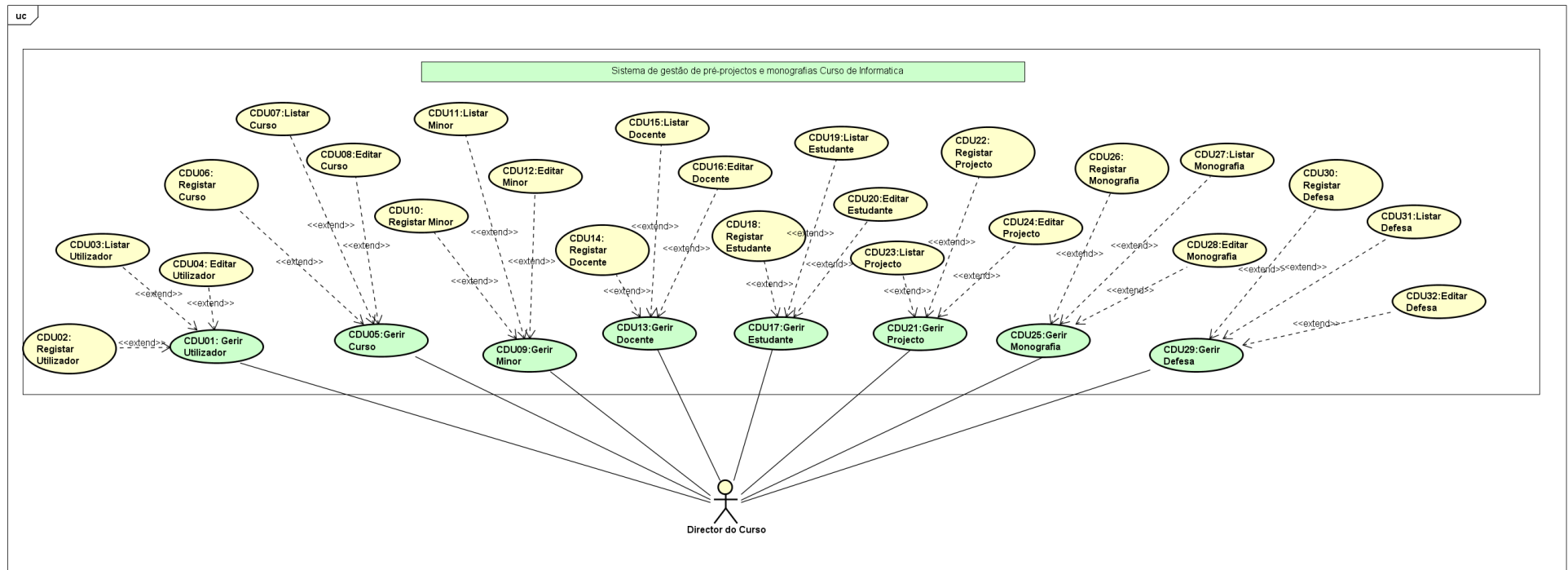


Figura 5: Representação do digrama de Casos de Uso

Fonte: Autor

B) Diagrama de Classe

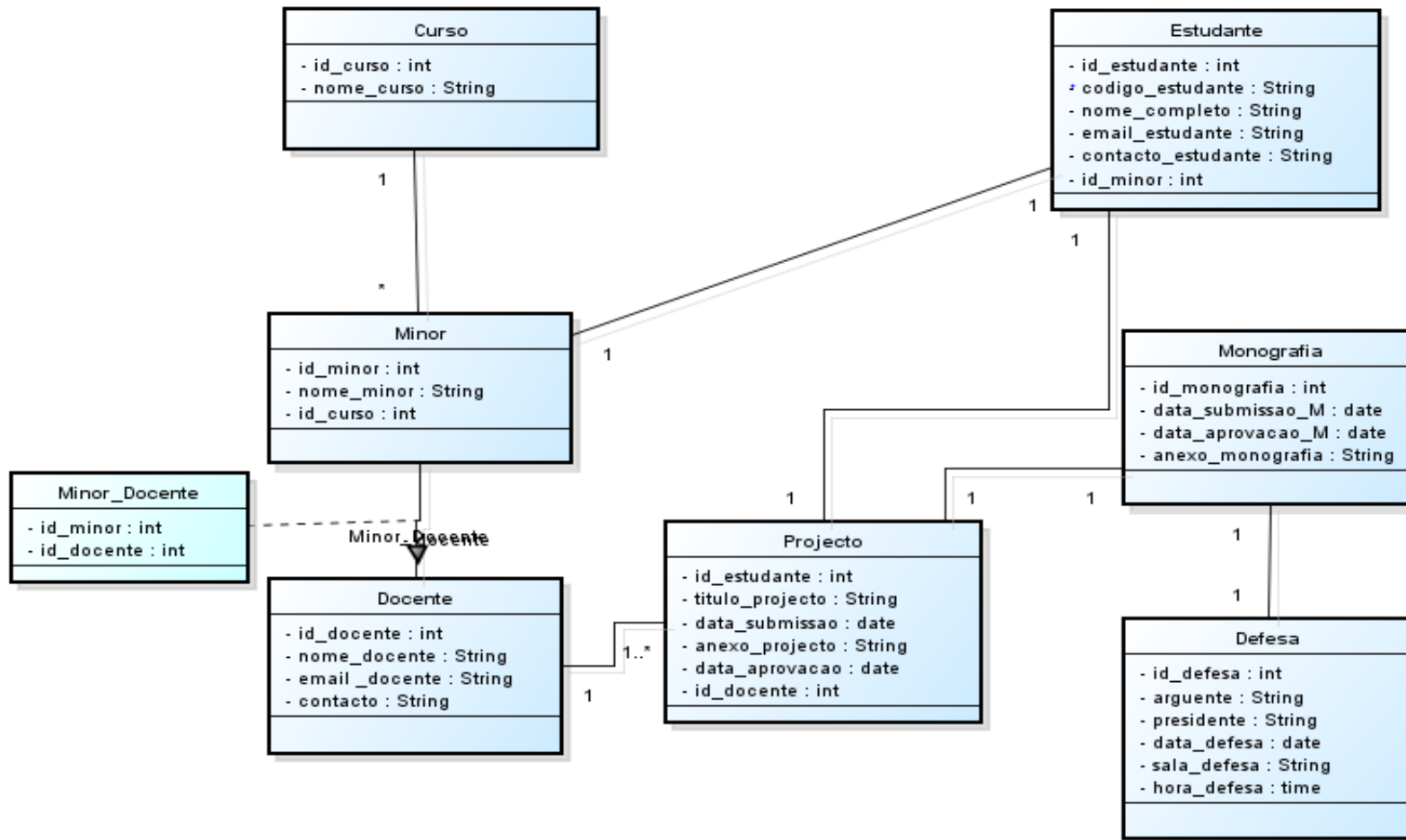


Figura 6: Representação do Diagrama de Classe
Fonte: Autor

C) Diagrama de Navegação

Pela necessidade de dar a transparência à fusão feita entre as duas metodologias apresentadas no referencial teórico (RUP e OOADM), o projecto navegacional corresponde a metodologia OOADM, onde e apresentado o diagrama de navegação do Sistema, com vista a apresentar o funcionamento detalhado do Sistema.

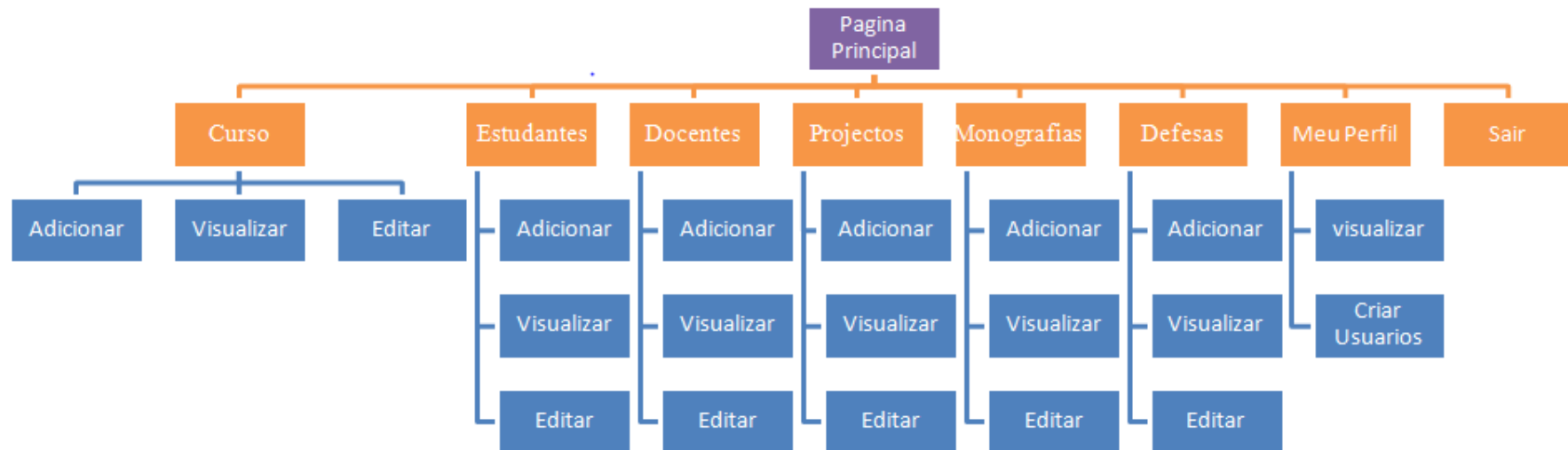


Figura 7: Representação do projecto navegacional (Director do curso).

Fonte: Autor

3.1.6. Projecto de Interface Abstracta

A figura 8 apresenta o painel principal da aplicação SGPM, que através deste painel o utilizador (chefe do Curso) tem acesso a todas funções do sistema. Sendo este resultado final da interacção entre o autor do sistema e o utilizador final, apresentando aspectos melhorados com relação a primeira versão apresentada ao utilizador, sendo este aprovado como modelo final do sistema.



Figura 8: Menu Principal do SGPM

Fonte: Autor

```

<section class="statistic">
  <div class="section_content section_content--p30">
    <div class="container-fluid">
      <p>O Sistema de Gestão de Pré-projectos & Monografias visa, melhorar aspectos de monitoria dos Pré-
      <div class="row">
        <div class="col-md-6 col-lg-3">
          <div class="statistic_item">
            <h2 class="number">{{estudantes}}</h2>
            <span class="desc">Estudantes Registados</span>
            <div class="icon">
              <i class="fa fa-users"></i>
            </div>
          </div>
        </div>
        <div class="col-md-6 col-lg-3">
          <div class="statistic_item">
            <h2 class="number">{{projectos}}</h2>
            <span class="desc">Numero de Projectos</span>
            <div class="icon">
              <i class="fa fa-book"></i>
            </div>
          </div>
        </div>
        <div class="col-md-6 col-lg-3">
          <div class="statistic_item">
            <h2 class="number">{{monografias}}</h2>
            <span class="desc">Numero Monografias</span>
            <div class="icon">

```

Figura 9: Ilustração de linhas de código do menu principal só SGPM

Fonte: Autor

```

        </div>
    </div>
    <div class="col-md-6 col-lg-3">
        <div class="statistic_item">
            <h2 class="number">{{ $monografias }}</h2>
            <span class="desc">Numero Monografias</span>
            <div class="icon">
                <i class="fa fa-book"></i>
            </div>
        </div>
    </div>
    </div>
    <div class="col-md-6 col-lg-3">
        <div class="statistic_item">
            <h2 class="number">{{ $defesas }}</h2>
            <span class="desc">Defesas agendadas</span>
            <div class="icon">
                <i class="fa fa-graduation-cap"></i>
            </div>
        </div>
    </div>
    </div>
</div>
</div>
</section>
<!-- END STATISTIC-->

```

Figura 10: Ilustração de linhas de código do menu principal só SGPM

Fonte: Autor











SGPM						☰
 <p>Super User Sair</p> <p>MENU PRINCIPAL</p> <ul style="list-style-type: none"> Curso Estudante Docente Projecto 	Esta em: Estudante				+ Registrar	
	ID	Código Do Estudante	Nome Completo	Minor	Accão	
	1	01.3904.2016	Luana Manuel	Engenharia em Desenvolvimento de Sistemas	  	
	2	01.0234.2016	liria mavimbe	Engenharia em Redes	  	
3	04.2402.2123	Felismina Lazaro	Engenharia em Multimedia	  		
Sistema de Gestão de Pré-projectos e Monografias, Universidade Pedagógica 2021						

Figura 11: Ilustração da tabela de dados do estudante.

Fonte: Autor

```

</thead>
<tbody>
  @foreach($estudantes as $key => $estudante)
    <tr>
      <td>{{ ++$key }}</td>
      <td>{{ $estudante->codigo_estudante }}</td>
      <td>{{ $estudante->nome_estudante }}</td>
      <td>{{ $estudante->minor->nome_minor }}</td>
      <td>
        <a class="btn btn-sm btn-info" href="{{ route('estudantes.show', $estudante->id) }}"
          class="fa fa-eye" title="Visualizar"></a>
        <a class="btn btn-sm btn-primary" href="{{ route('estudantes.edit', $estudante->id)
          class="zmdi zmdi-edit" title="Atualizar"></a>
        {!! Form::open(['method' => 'DELETE', 'route' => ['estudantes.destroy',
          $estudante->id], 'style'=>'display:inline']) !!}
        {!! Form::button('<i class="zmdi zmdi-delete"></i>', ['type' => 'submit', 'class' =>
          {!! Form::close() !!}
      </td>
    </tr>
  @endforeach
</tbody>
</table>
</div>
<!-- END DATA TABLE-->
</div>

```

Figura 12: Ilustração do código da tabela com dados do estudante

Fonte: Autor

```

</thead>
<tbody>
  @foreach($estudantes as $key => $estudante)
    <tr>
      <td>{{ ++$key }}</td>
      <td>{{ $estudante->codigo_estudante }}</td>
      <td>{{ $estudante->nome_estudante }}</td>
      <td>{{ $estudante->minor->nome_minor }}</td>
      <td>
        <a class="btn btn-sm btn-info" href="{{ route('estudantes.show', $estudante->id) }}"
          class="fa fa-eye" title="Visualizar"></a>
        <a class="btn btn-sm btn-primary" href="{{ route('estudantes.edit', $estudante->id)
          class="zmdi zmdi-edit" title="Atualizar"></a>
        {!! Form::open(['method' => 'DELETE', 'route' => ['estudantes.destroy',
          $estudante->id], 'style'=>'display:inline']) !!}
        {!! Form::button('<i class="zmdi zmdi-delete"></i>', ['type' => 'submit', 'class' =>
          {!! Form::close() !!}
      </td>
    </tr>
  @endforeach
</tbody>
</table>
</div>
<!-- END DATA TABLE-->
</div>

```

Figura 13: Ilustração da tabela de dados do estudante

Fonte: Autor

Figura 14: Ilustração da tela de registo de estudante.
Fonte: Autor

```

<div class="modal-body">
  {!! Form::open(array('route' => 'estudantes.store', 'method'=>'POST')) !!}
  @csrf()

  <div class="col-lg-12">
    <div class="card">
      <div class="card-header">
        <strong>Novo Estudante</strong>
      </div>
      <div class="card-body card-block">
        <div class="form-group">...
        </div>
        <div class="form-group">
          <strong>Nome :</strong>
          {!! Form::text('nome_estudante', null, array('placeholder' => 'nome','class' =>
            'form-control')) !!}
        </div>
        <div class="form-group">
          <strong>E-mail:</strong>
          {!! Form::text('email_estudante', null, array('placeholder' => 'e-mail','class'
            =>
            'form-control')) !!}
        </div>
        <div class="form-group">
          <strong>Contacto</strong>
          {!! Form::text('contacto_estudante', null, array('placeholder' =>
            'contacto','class' =>
            'form-control')) !!}
        </div>
      </div>
    </div>
  </div>

```

Figura 15: Ilustração de código de registo de estudante
Fonte: Autor

```

>
<div class="form-group"> ...
</div>
</div>
</div>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Cancelar</button>
<button type="submit" class="btn btn-primary">Salvar</button>
</div>
{!! Form::close() !!}
</div>
</div>
</div>
<!-- end modal curso -->
</div>
@endsection

```

Figura 16: Ilustração de código de registo de estudante

Fonte: Autor

3.2.Implementação

Nesta fase de implementação do sistema, são apresentados todos os recursos e ferramentas necessárias para a instalação do SGPM.

Acessos ao servidor:

Recursos de hardware:

- ✓ Placa de rede com acesso a internet;
- ✓ 4GB de memoria RAM;
- ✓ 100GB de espaço de disco livre;

Recursos mínimos para o software:

- ✓ MySQL, PHP;
- ✓ Navegador;
- ✓ Apache2 ou qualquer web Server com os mesmos recursos.
- ✓ Recursos para instalação e configuração:
- ✓ Editor de texto para HTML, PHP, JS, entre vários (VS Code, Sublime Text Editor, Notepad++);
- ✓ Descompactador de arquivos de ZIP/RAR (WinRAR, Flzip, WinZip);
- ✓ Sistema de Gestão de Base de Dados (PHPMYAdimin);
- ✓ Cliente SSH, (Putty, SmarTTY) para instalações remotas;

Domínio: www.sgpm.up.ac.mz

3.3. Validação dos objectivos da pesquisa

Objectivo Geral - Melhorar o processo de controle das Monografias através de um sistema de Gestão de Informação – Após o desenvolvimento desta pesquisa, é notório que foram apresentadas melhorias no que concerne ao fornecimento de dados quantitativos e qualitativos no controlo de estudantes inscritos no departamento.

- ✓ Analisar o processo actual de controle de Monografias no Curso de Informática – O processo anteriormente usado foi avaliado e foram extraídos os pontos fracos que foram melhorados no novo processo de controle e tratamento dos pré-projectos e monografias.
- ✓ Integrar as Normas e políticas estabelecidas no Regulamento da Universidade Pedagógica, de modo a serem cumpridas – foram integradas normas da instituição para o processo de culminação do curso, deste a submissão do pré-projecto a validação;
- ✓ Implementar Mecanismos de segurança que possam garantir a integridade de dados no sistema – no âmbito do desenvolvimento do sistema, foram implementados mecanismos de segurança para garantir a integridade da informação: só tem acesso a informação gerada pelo sistema quem for devidamente autenticado pelo sistema, a partir de log que exige credenciais utilizador inscrito e a palavra-se, e também medidas de segurança do Framework de desenvolvimento, para garantir integridade e confidencialidade da informação que circula dentro do sistema;
- ✓ Implementar técnicas para a melhoria de tomada de decisão, no âmbito de alocação dos projectos – com o sistema em funcionamento o curso de informática passa a ter melhor visibilidade e controle do que acontece em torno dos projectos e alocações, ou seja, o sistema fornece estáticas que ajudam na tomada de decisões e no controle de estudantes, projectos e docentes relacionados aos mesmo, desta forma o curso mantém-se abalizado das actividades dos docentes do departamento.

CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES

4.1.Considerações finais

Terminado o estudo sobre o tema em causa, compete-nos referir que vários aspectos referentes ao processo de gestão foram estudados e analisados de forma integrada e profunda, que com base nisto, foi possível entender a regra de negocio e o desenvolvimento de uma possível solução.

Durante o estudo, foi possível identificar as possíveis lacunas que poderiam, de alguma forma, fazer com que a informação não fluísse da forma mais ideal e esperada, pois havia morosidade no acesso a informação, na resposta para os utentes e também na organização e gestão por parte do Curso de Informática, sendo que, esta tem tantas outras diversificadas tarefas para além da gestão de Pré-projectos e Monografias. O estudo teve como foco dinamizar o processo de gestão de Pré-projectos e Monografias usando bases tecnológicas especificamente, soluções Web.

A construção da solução para pesquisa em causa culminou com o desenvolvimento de um sistema de gestão informatizado, que para o seu processo, contou com a fusão de duas metodologias escolhidas pela Autora, pois ao longo do estudo foi possível perceber que para o desenvolvimento de soluções Web, as metodologias podem ser aplicadas de forma conjunta, com o objectivo de ter mais qualidade no produto final e tratar na integra todos os pontos necessários.

A Implementação da solução foi facilitada, tendo em conta que o Curso de Informática possui uma estrutura para receber este tipo de soluções. Como é sabido, as soluções Web são muito bem conhecidas pela sociedade no geral tanto que para integração desta no espaço académico da UPM, não enfrentou-se muitas dificuldades.

Diante disto, conotamos que a pesquisa foi concluída de forma satisfatória, pois foram atendidos todos objectivos traçados desde o início da pesquisa, bem como as regras do negocio foram aplicadas de forma a obter resultados funcionais e exequíveis.

Para o desenvolvimento futuro, o Autor recomenda que, se tratando de um sistema que pode ser aplicado para vários cursos da instituição, seja implementado de acordo com a necessidade do departamento em causa, otimizando o processo actual e criando melhorias no desempenho do sistema atendendo a altas demandas.

5. Referências bibliográficas

- ALVES, W. P. *Fundamentos de bancos de dados*. São Paulo, SP: Érica, 2005.
- BABA, F.H. *Modelagens de aplicação multimídia*. UEL, 2005.
- BIO, Sérgio Rodrigues. *Sistemas de informação: um enfoque empresarial*. São Paulo: Atlas, 1985.
- BRITO, L. S. F. "WEBSCHARTS: Uma Ferramenta de desenvolvimento de aplicações Web baseada no HMBS/M", Dissertação de Mestrado, UFMS, 2003.
- CARVALHO, A. M. B. R., CHIOSSI, T. C. S. *Introdução à Engenharia de Software*. Editora Unicamp, 2001.
- CORNACHIONE Jr., Edgard B. *Informática aplicada às áreas de contabilidade, administração e economia*. São Paulo: Atlas, 1998.
- CUNHA, M.C.R. *Autoria em Hipermídia: O Modelo OOHDM Aplicado ao Ensino de Linguagens de Programação*. UFLA, 2002.
- DE ABREU; B. *Desenvolvimento de um sistema Web para utilização e gerenciamento de dados de Cupons Fiscais e Saúde*. Universidade Federal de Ouro Preto – UFOP, Brasil, 2016.
- FERREIRA, A. *Projecto de aplicação WEB por meio da linguagem webml*. UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR, Brasil, 2013.
- FRANÇA, R.N. *Desenvolvimento de um Sistema Hipermídia Educacional Aplicado a Anatomia Veterinária*. UFLA, 2005.
- GIL, António Carlos. *Métodos e técnicas de pesquisa social*. 5.ed. São Paulo: Atlas, 1999.
- LAKATOS, Eva Maria. *Fundamentos da metodologia científica/ Marina de Andrade Marconi - 5ª Ed.* São Paulo: Atlas, 2003.
- LAUDON, Kenneth C.; LAUDON, Jane Price. *Gerenciamento de sistemas de informação*. Rio de Janeiro: LTC, 1990.
- LAUDON, Kenneth C.; LAUDON, Jane P. *Sistemas de informação*. Rio de Janeiro: LTC, 2014.
- LIMA, F. *Modelagem Semântica de Aplicações WWW*. 128p. Tese de Doutorado em Ciência da Computação - PUC. Rio de Janeiro – RJ, 2003.

- KASPER, H. *O processo de pensamento sistêmico: Um estudo das principais abordagens a partir de um quadro de referência proposto*. UFRGS, Brasil, 2000.
- MARTIN, G.; MCCLURE, C. *The CASE Experience*. Byte, 1991.
- NIXON, Robin. *Learning PHP, MySQL, JavaScript, and CSS*. 2. ed. United States of America, 2012.
- O'BRIEN, James A. *Sistemas de informação*. São Paulo: Saraiva, 2000
- PADOVEZE, Clóvis Luís. *Contabilidade gerencial: um enfoque e sistemas de informação contábil*. São Paulo: Atlas, 1997.
- PEREIRA, R.S. *Sistema Multimídia Para Ensino e Aprendizado de Irrigação*. UFV, 2006.
- PRESSMAN, Roger. *Engenharia de Software*. 6ª ed. São Paulo: McGraw-Hill, 2006.
- PRODANOV, C. C.; FREITAS, E. C. *Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico*. 2. ed. Novo Hamburgo: Feevale, 2013.
- RAMOS, Denis. *Ambientes de desenvolvimento de software (ads) & ferramentas case: importância e aplicações*. Universidade Federal de Minas Gerais, Brasil, 2011.
- ROSA, Maria Virgínia de Figueiredo Pereira do Couto; ARNOLDI, Marlene Aparecida Gonzalez Colombo. *A entrevista na pesquisa qualitativa: mecanismos para a validação dos resultados*. Belo Horizonte: Autêntica Editora, 2006.
- ROSSI, J.M. *Para uma tecnologia educacional*. Porto Alegre: Editora ArtMed, 1999.
- SCHWABE, D., ROSSI, G. *From domain models to hypermedia applications: an object-oriented approach*. In: International workshop on methodologies for designing hypermedia applications, 1994.
- SCHWABE, D., ROSSI, G. *Systematic Development of Hypermedia Applications Using OOHDM*. Departamento de Informática, PUC-Rio, 1999.
- SCHWABE, D.; ROSSI, G. *Notação do modelo OOHDM*. Rio de Janeiro, 2002.
- SCHWABE, Daniel; PONTES, Rita; MOURA, Isabela. *OOHDM-Web: An Environment for Implementation of Hypermedia Applications in the www*. 1994.
- SCHWABE, Daniel; ROSSI, Gustavo. *The Object-Oriented Hypermedia Design Model*. Brazil, 2014.
- SEBASTA, Robert W. *Conceitos de linguagens de programação*. 4. ed. Tradução José Carlos Barbosa dos Santos. Porto Alegre: Bookman, 2000.

- SEMEDO; Maria Joao. *Ganhos de produtividade e de sucesso de Metodologias Ágeis VS Metodologias em Cascata no desenvolvimento de projectos de software*. Universidade Lusófona de Humanidades e Tecnologias, Lisboa. 2012.
- SILVA, Alberto; VIDEIRA, Carlos: *UML, Metodologias e Ferramentas CASE*. Porto-Lisboa, 2001.
- SILVA, H.R. *Desenvolvimento de um Aplicativo Hipermedia Educacional para Sistemas de Cabeamento Estruturado*. UFLA, 2012.
- SOMMERVILLE, Ian. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.
- SOMMERVILLE, Ian. *Software Engineering*. Fifth Edition. Addison-Wesley, 1995.
- SUEHRING, Steve. *MySQL, A Bíblia*. 1 ed. Rio de Janeiro: Campus, 2002.
- WINCKLER, M.; Farenc, C.; Palanque, P.; Bastide, R. Designing Navigation for Web Interfaces. In Proceedings:... IHM-HCI2001, Lille, França, 10-14 Sep. 2001.
- ZAMBALDE, A. L., et al.: “*Modelagem, autoria e análise de usabilidade de aplicação hipermedia direcionada ao setor agropecuário*”, UFLA, 1999.

Anexos



FACULDADE DE ENGENHARIAS E TECNOLOGIAS
CURSO DE INFORMÁTICA

Campus da Lhanguene, Av de Trabalho, 24821, Maputo Tel: +258 22 41880



CALENDÁRIO DE DEFESAS DE MONOGRAFIAS CIENTÍFICAS

Nº Ordem	Nome do Candidato	Tema	Júri	Horas	Local
Data	21.04.2021	LICENCIATURA DE INFORMÁTICA			
1	Suale Abdul Suale	O Contributo de Sistemas de Informação na Gestão Hospitalar - Caso do Centro Médico Privado do Benfica	Presidente: dr. Ricardo Uainda Supervisor: Professor Doutor Félix Singo Arguente: Mestre Célio Sengo	10:00	Inf. C
2	Amina Geraldo Jequecene	Implementação de tecnologia Push To Talk over Cellular (SmartPhone) como forma de melhoria na comunicação com Aparelhos Push to Talk – caso de estudo: Eletricidade de Moçambique.	Presidente: Mestre Valdinacio Florêncio Paulo Supervisor: dr. Xadrique Agnesio Macamo Arguente: dr. Xavier Bila	11:30	Inf. C
3	Adriana Geraldo Jequecene	Implementação do Método Engenharia Web Baseada em UML na Disseminação de uma Aplicação Web para a Gestão de Vigilantes - Caso de Estudo: Comissão dos Exames de Admissão	Presidente: Mestre Sheila António Siteo Supervisor: Mestre Célio Sengo Arguente: Mestre Cláudia Ivete Jovo Gune	13:00	Inf.C

Director do Curso

Maputo, 15 de Abril de 2021

Chefe do Departamento Pedagógico

Sheila Antonio Siteo

(Mestre Sheila Antonio Siteo)

Cláudia Jovo Gune

(Mestre Cláudia Ivete F. Jovo Gune)

Calendário de defesa produzido manualmente

Apêndices



FACULDADE DE ENGENHARIAS E TECNOLOGIAS
CURSO DE INFORMATICA

Campus do Lhanguene, Av Do Trabalho, 24821, Maputo Tel: +258 22 41880

VISTO

A Directora Adjunta de Graduacao
(Mestre Evelina C.Sambane)

CALENDARIO DE DEFESAS DE MONOGRAFIAS CIENTIFICAS

N Ordem	Nome Candidato	Titulo da Monografia	Juri	Horas	Data Local
Data	2021-12-15	LICENCIATURA EM INFORMATICA			
1	liria mavimbe	Sistema de Manutencao de dados	Presidente: Celio Sengo Supervisor: Celio Sengo Arguente: Felicidade Julia	11:58:00	INFO D

Director do Curso
(Mestre Sheila Antonio Siteo)

Maputo, 02-12-2021

Chefe do Departamento Pedagogico
(Mestre Claudia Ivete F.Jovo Gune)

Figura 17: Calendário de defesas produzido pelo sistema

Fonte: Autor