

Universidade do Minho
Escola de Engenharia

Vanilde Lima da Silva

**Utilização de técnicas de Gestão de
Projetos na análise de requisitos de
projetos de *software***



Universidade do Minho

Escola de Engenharia

Vanilde Lima da Silva

**Utilização de técnicas de Gestão de
Projetos na análise de requisitos de
projetos de *software***

Dissertação de Mestrado
Mestrado em Sistemas de Informação

Trabalho realizado sob a orientação do
Professor Doutor Ricardo J. Machado

Julho de 2012

DECLARAÇÃO

Nome: VANILDE LIMA DA SILVA

Endereço Electrónico: vanilde.lima@gmail.com

N.º. do Bilhete de Identidade: OL838K072

Título da Dissertação de Mestrado:

Utilização de técnicas de Gestão de Projetos na análise de requisitos de projetos de software.

Orientador: Professor Doutor Ricardo J. Machado

Ano da conclusão: 2012

Designação do Mestrado: Mestrado em Sistemas de Informação

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA DISSERTAÇÃO APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE.

Universidade do Minho, ____ / ____ / _____

Assinatura: _____

Agradecimentos

Primeiramente a Deus por me abençoar em todos os momentos da minha vida, fazendo-se presente nas horas mais difíceis, acolhendo-me em Seus braços e dando-me forças para prosseguir.

Aos meus pais que sempre estiveram presentes na minha educação me apoiando e instruindo-me para a importância de estar sempre adquirindo novos conhecimentos e trabalhando para o alcance dos meus objetivos. Por toda energia transportada apesar da distância e da saudade.

Ao meu amado esposo, pelo apoio, carinho, incentivo, compreensão nas horas de ausência, enfim, por toda força e confiança transmitidas que, com certeza, foram fatores muito importantes para realização deste trabalho.

À minha querida filha, pelo seu amor e carinho incondicional, pois, apesar de sua pouca idade, soube compreender os vários momentos em que não lhe foi possível dar atenção, em que mais precisei dedicar-me ao trabalho. Por existir em minha vida e me proporcionar dias eternos de ternura e felicidade.

Ao meu orientador, Prof. Dr. Ricardo J. Machado, pela disposição e auxílio em todas as etapas, sempre com total atenção e dedicação. Pelas orientações e profissionalismo durante todo o desenvolvimento deste trabalho.

Aos meus irmãos e amigos, que souberam compreender a minha falta de comparência, encorajando-me para que eu prosseguisse em busca do sucesso.

E, por fim, a todos que de uma forma indireta participaram deste período de grande importância em minha vida.

Resumo

A falta de efetividade no processo de levantamento de requisitos tem resultado em sistemas que não atendem às necessidades do cliente ou, ainda, produtos que apresentam baixa qualidade. Diante deste contexto, em que se evidenciam grandes dificuldades em conciliar os requisitos de um sistema com o produto final efetivamente entregue, as organizações têm buscado apoio em vários recursos tecnológicos e metodologias, visando garantir a qualidade no desenvolvimento de seus produtos e serviços. A área de Gestão de Projetos, por sua vez, tem demonstrado grande maturidade e, neste sentido, suas melhores práticas estão compiladas num único guia, qual seja, o *Project Management Body of Knowledge* – PMBoK, editado pelo *Project Management Institute* – PMI, de amplo uso. A proposta do presente trabalho é estudar a aplicabilidade de algumas das técnicas, típicas para o delineamento de âmbito em Gestão de Projetos, no levantamento de requisitos em desenvolvimento de sistemas orientados a objetos. Desta forma, apresentam-se, neste trabalho, os principais artefatos relacionados ao levantamento de requisitos de sistemas orientados a objetos, de acordo com a metodologia *Rational Unified Process* - RUP. Igualmente, discutem-se alguns conceitos e a relevância da disciplina de Gestão de Projetos, com destaque para a abordagem contida no guia PMBoK, especialmente no que diz respeito às técnicas sugeridas para a definição de âmbito. A partir destes entendimentos aprofundados, foi efetuada uma análise da similaridade conceitual entre aquelas áreas relativamente ao levantamento de requisitos, cujo resultado foi a proposta de um processo que permite o uso de técnicas de definição de âmbito no universo de desenvolvimento de sistemas orientados a objetos. Desta forma, para se experimentar a viabilidade prática da proposta, foi efetuado um estudo de caso sobre um sistema real desenvolvido segundo a abordagem de orientação a objetos. Por fim, vantagens e desvantagens do processo de proposto foram apresentadas e discutidas.

Palavras-chave: Requisitos, Desenvolvimento de Sistemas, Projeto, PMBoK, RUP, UML.

Abstract

The lack of effectiveness in the requirements gathering process has resulted in systems that do not meet customer needs, or in poor quality products. In this light, while tackling great difficulties in accommodating a system's requirements with the final product effectively delivered, organizations have sought after support in various technological resources and methodologies, in order to ensure quality in the development of their products and services. On the other hand, the Project Management field has evolved and matured immensely. In this sense, its best practices are compiled in a handbook under the name Project Management Body of Knowledge - PMBoK, edited by the Project Management Institute – PMI, and which is widely used. This work' scope addresses the applicability of some of its techniques which are common within the project management scope design, whilst identifying object-oriented systems' requirements. Under the aforementioned circumstances, this study seeks to display the main artifacts related to the object-oriented systems requirements, in accordance with the methodology known as Rational Unified Process - RUP. Some concepts deemed relevant for the project management discipline are thoroughly analyzed in line with the PMBoK Handbook approach, especially in what regards the definition of scope techniques. Whilst resting upon such in-depth understanding, it was made an analysis of the conceptual similarities between those areas, concerning requirements gathering, which resulted in a proposal of a process that allows the use of scope definition techniques in the object-oriented systems development. In this manner, to test the feasibility of such proposal, a case study was conducted over an actual system developed according to the object-oriented approach. Finally, advantages and disadvantages of the proposed process were presented and discussed at length.

Keywords: Requirements, Systems Development, Project, PMBoK, RUP, UML.

Índice

| | |
|--|------|
| Resumo | v |
| Abstract | vii |
| Índice | ix |
| Índice de figuras | xiii |
| Índice de Tabelas | xv |
| Lista de Siglas | xvii |
| 1 Introdução | 1 |
| 1.1 A importância de uma boa definição de requisitos | 1 |
| 1.2 Objetivos do estudo | 6 |
| 1.3 Abordagem metodológica | 7 |
| 1.4 Estrutura do documento | 9 |
| 2. Conceitos relevantes em Análise de Sistemas | 11 |
| 2.1 Introdução | 11 |
| 2.2 Documentos em Análise Estruturada e Essencial | 12 |
| 2.3 Artefatos em Análise Orientada a Objetos | 15 |
| 2.4 Conclusão | 26 |
| 3. Conceitos relevantes em Gestão de Projetos | 29 |
| 3.1 Introdução | 29 |
| 3.2 Termo de Abertura do Projeto | 31 |
| 3.3 Declaração de Âmbito | 32 |
| 3.4 <i>Work Breakdown Structure – WBS</i> | 34 |
| 3.5 Matriz de Responsabilidades | 36 |
| 3.6 Diagrama de Precedência | 38 |
| 3.7 Conclusão | 39 |

| | |
|---|----|
| 4. Análise de similaridades conceituais e proposta de processo para uso de documentos de Gestão de Projetos no delineamento de requisitos em AOO..... | 41 |
| 4.1 Introdução..... | 41 |
| 4.2 Análise de similaridades conceituais | 41 |
| 4.3 Processo proposto | 47 |
| 4.3.1 Elaboração da Declaração de Âmbito..... | 48 |
| 4.3.2 Elaboração da WBS..... | 48 |
| 4.3.3 Elaboração da Matriz de Responsabilidades..... | 49 |
| 4.3.4 Elaboração dos Diagramas de Casos de Uso iniciais | 51 |
| 4.3.5 Elaboração do Diagrama de Classes inicial | 53 |
| 4.4 Conclusão | 54 |
| 5. Estudo de Caso..... | 57 |
| 5.1 Considerações iniciais..... | 57 |
| 5.2 Sobre a instituição e o sistema SIAQ-SCMG..... | 57 |
| 5.3 Aplicação do processo proposto..... | 59 |
| 5.3.1 Elaboração da Declaração de Âmbito..... | 59 |
| 5.3.2 Elaboração do WBS..... | 61 |
| 5.3.3 Elaboração da Matriz de Responsabilidades..... | 66 |
| 5.3.4 Elaboração dos Casos de Uso iniciais | 68 |
| 5.3.5 Elaboração do Diagrama de Classes inicial | 71 |
| 5.4 Considerações sobre o processo..... | 72 |
| 6. Conclusões e trabalhos futuros..... | 75 |
| Referências..... | 81 |
| ANEXOS | 87 |
| Anexo 1 - WBS elaborada a partir dos requisitos funcionais da documentação original..... | 89 |
| Anexo 2 - WBS obtida após aplicação da técnica proposta | 90 |

| | |
|---|----|
| Anexo 3 - Matriz de responsabilidades elaborada a partir dos requisitos funcionais da documentação original | 91 |
| Anexo 4 - Matriz de responsabilidades decorrente da aplicação da proposta..... | 92 |

Índice de figuras

| | |
|---|----|
| Figura 1 - Resultados dos estudos do <i>Standish Group</i> | 4 |
| Figura 2 - Triângulo de restrições do projeto | 5 |
| Figura 3 - Fases da abordagem metodológica em curso..... | 8 |
| Figura 4 - Desenvolvimento de Sistema pelo Método da Análise Estruturada | 12 |
| Figura 5 - Exemplo de DFD típico em Análise Estruturada | 13 |
| Figura 6 - O Modelo Essencial | 14 |
| Figura 7 - Diagrama de Caso de Uso | 17 |
| Figura 8 - Diagrama de Estados..... | 18 |
| Figura 9 - Diagrama de atividades | 18 |
| Figura 10 - Diagrama de Classes..... | 19 |
| Figura 11 - Diagrama de Objetos | 20 |
| Figura 12 - Diagrama de Sequência..... | 21 |
| Figura 13 - Diagrama de Colaboração/Comunicação | 21 |
| Figura 14 - Diagrama de Componentes | 22 |
| Figura 15 - Diagrama de Implantação..... | 22 |
| Figura 16 - Relacionamento entre fases e disciplinas segundo o IBM RUP..... | 23 |
| Figura 17 - As fases e os marcos de um projeto | 24 |
| Figura 18 - Artefatos das disciplinas (a) <i>Modelagem de Negócios</i> e (b) <i>Requisitos</i> , respectivamente..... | 25 |
| Figura 19 - Artefatos da disciplina "Análise e <i>Design</i> " | 26 |
| Figura 20 - Relacionamento entre as fases em gestão de projetos, segundo o PMBoK..... | 30 |
| Figura 21 - Processos sobrepostos de gestão de projetos..... | 30 |
| Figura 22 - Hierarquia de termos de uma WBS..... | 35 |
| Figura 23 - Exemplo de decomposição com principais entregas | 36 |
| Figura 24 - Método do Diagrama de Precedência..... | 38 |
| Figura 25 - Estratégia geral de análise de similaridades, vista a partir das fases de cada dimensão | 42 |
| Figura 26 - Análise de similaridades conceituais entre Gestão de Projetos e AOO | 43 |
| Figura 27 - Estratégia inicial de conexões conceituais entre documentos e artefatos..... | 45 |
| Figura 28 - Estratégia final de conexões conceituais entre documentos e artefatos | 46 |
| Figura 29 - Exemplo de trecho de WBS para um Sistema de Gestão de Biblioteca..... | 49 |

| | |
|--|----|
| Figura 30 - Esquema de elaboração da Matriz de Responsabilidades, a partir da WBS e da Declaração de Âmbito..... | 50 |
| Figura 31 - Exemplo de parte de uma Matriz de Responsabilidades elaborada a partir de uma WBS..... | 51 |
| Figura 32 - Esquema de elaboração de casos de uso, a partir da Matriz de Responsabilidades..... | 52 |
| Figura 33 - Exemplo de caso de uso derivado de Matriz de Responsabilidades | 52 |
| Figura 34 - Esquema de elaboração de classes e seus métodos a partir da Matriz de Responsabilidades..... | 53 |
| Figura 35 - Diagrama de Classes parcial derivado de uma Matriz de Responsabilidades..... | 54 |
| Figura 36 - Organograma da Santa Casa de Misericórdia de Guimarães (SCMG) | 58 |
| Figura 37 - WBS elaborada a partir dos requisitos funcionais do SIAQ - SCMG..... | 65 |
| Figura 38 - Caso de uso "Gerir utentes" | 70 |
| Figura 39 - Diagrama de Classes inicial para "2. Gerir Utentes" | 72 |

Índice de Tabelas

| | |
|--|----|
| Tabela 1 - Resumo das abordagens metodológicas | 15 |
| Tabela 2 - Matriz de responsabilidades | 37 |
| Tabela 3 - Relação entre documentos e artefatos, a partir das questões centrais | 44 |
| Tabela 4 - Resumo das adequações propostas aos documentos de Gestão de Projetos | 56 |
| Tabela 5 - Declaração de âmbito sugerida para o Estudo de Caso..... | 62 |
| Tabela 6 - Requisitos funcionais do SIAQ – SCMG..... | 66 |
| Tabela 7 - Visão parcial da matriz de responsabilidades SIAQ - SCMG | 69 |
| Tabela 8 - Visão parcial da matriz de responsabilidades – Tema/assunto " 2. Gerir utentes" | 70 |

Lista de Siglas

| | |
|---------|--|
| AOO | Análise Orientada a Objetos |
| CEO | <i>Chief Information Officer</i> |
| CIO | <i>Chief Exexutive Officer</i> |
| DFD | Diagrama de Fluxo de Dados |
| DER | Diagrama de Entidades e Relacionamentos |
| EAP | Estrutura Analítica do Projeto |
| IBM RUP | <i>IBM Rational Unified Process</i> |
| IT | <i>Information Technology</i> |
| ITGI | <i>Information Technology Governance Institute</i> |
| IPMA | <i>International Project Management Association</i> |
| JAD | <i>Joint Application Design</i> |
| MR | Matriz de Responsabilidades |
| NDLTD | <i>Networked Digital Library of Theses and Dissertations</i> |
| OMG | <i>Object Management Group</i> |
| OMT | <i>Object Modeling Technique</i> |
| OO | Orientado a Objeto |
| OOSE | <i>Object Oriented Software Engineering</i> |
| PMBok | <i>Project Management Body of Knowledge</i> |
| PMI | <i>Project Management Institute</i> |
| PM4DEV | <i>Project Management for Development Organizations</i> |
| PRINCE2 | <i>PRojects IN Controlled Environments</i> |
| SADT | <i>Structured Analysis and Design Technique</i> |
| SIAQ | Sistema de Apoio à Qualidade |
| SCMG | Santa Casa de Misericórdia |
| TAP | Termo de abertura do projeto |
| TCS | <i>Tata Consultancy Services</i> |
| TI | Tecnologia da Informação |
| UML | <i>Unified Modeling Language</i> |
| XP | <i>EXtreme Programming</i> |
| WBS | <i>Work Breakdown Structure</i> |
| ZOPP | <i>Ziel Orientierte Projekt Planung</i> |

1 Introdução

1.1 A importância de uma boa definição de requisitos

Com o aumento da concorrência e clientes cada vez mais exigentes, as empresas têm procurado constantemente aperfeiçoar seus conhecimentos para garantir a qualidade de seus produtos e serviços. Isso faz com que um alto investimento em novas tecnologias seja despendido com o objetivo de que tais adequações lhes proporcionem uma potencial vantagem competitiva. Efetivamente, conforme informado por pesquisa realizada pelo ITGI- *IT Governance Institute*, para compor o Relatório Global de Governança de TI, de 2008, cerca de 93% dos pesquisados (CIO e CEO em 23 países) consideram que a TI tem importância de média a alta para a estratégia corporativa geral [IT Web, 2008].

Dentre estes investimentos, inclui-se a obtenção de sistemas informáticos ou aplicações informáticas, entendidos como sendo “um conjunto de componentes inter-relacionados que coletam, manipulam, armazenam e divulgam dados e informações e, ainda, fornecem um mecanismo de *feedback* para atender um objetivo. Tal mecanismo de *feedback* ajuda as organizações a alcançarem suas metas, aumentar seus lucros ou melhorar os serviços oferecidos aos clientes. As empresas podem utilizar sistemas de informação para aumentar as receitas e reduzir custos” [Stair e Reynolds 2008, p.4].

Porém, nem todos esses investimentos em tecnologia trazem a estas empresas os benefícios almeçados [Martinsons e Chong, 1999; Standing, Guilfoyle, Lin e Love, 2006; Avgerou e McGrath, 2007]. Em consonância com o entendimento desses últimos autores, Dalmazo [2008] também cita um relatório produzido pela *Tata Consultancy Services* (TCS), onde consta que um em cada três projetos de TI não atinge às expectativas dos contratantes. Ao se tentar compreender as causas de tantas falhas, observa-se que, particularmente no âmbito do desenvolvimento de sistemas, uma tarefa crítica, que por vezes não é adequadamente realizada, é a de se definir o âmbito do sistema, através da caracterização de suas funcionalidades [Silva e Videira 2001, p.19].

Os autores Aurum e Wohlin [2005], Mecnas e Oliveira [2005] e Sommerville [2007], advertem acerca da importância da fase de identificação ou levantamento de requisitos no processo de desenvolvimento de aplicações informáticas. De acordo com esses autores, para que existam maiores possibilidades de o sistema passar por todas as etapas do seu ciclo de vida e chegar a sua conclusão com sucesso, faz-se necessário que esta etapa inicial seja bem realizada para que os erros então detectados não alcancem as etapas seguintes.

De fato, parece haver uma dificuldade em se compreender, com clareza, as necessidades do cliente quando do levantamento dos requisitos, os quais são descritos por Larman [2008, p. 81] como “as capacidades e condições às quais o sistema e, em termos mais amplos, o projeto, deve atender”. Em consonância com Larman [2008], para Kulak e Guiney [2001, p.4], um requisito é “algo que uma aplicação de computador deve fazer para os seus utilizadores. Essa é uma função ou característica ou princípio específico que o sistema deve prover. Os requisitos constituem o âmbito do projeto de desenvolvimento do sistema”. De acordo com Jacobson, Booch e Rumbaugh [1999], o processo de desenvolvimento de sistemas trata-se de um conjunto completo das atividades que se fazem necessárias para transformação dos requisitos levantados em artefatos que representem um sistema.

Segundo Sommerville [2007], as descrições das funções e restrições do que um sistema deverá realizar são os seus requisitos. Estes, considerados em conjunto e interrelacionados, irão definir a estrutura e os comportamentos necessários ao funcionamento do sistema. Ainda para o autor, tratam-se de itens de um requisito: os dados, processos, restrições de acordo com o negócio, as pessoas envolvidas e o relacionamento entre todos esses elementos. Pressman [1995], por sua vez, tinha um entendimento que se faz presente mesmo nos dias atuais, pois afirmava que o levantamento de requisitos deveria ser de forma *top-down*. Deste modo, sugeria o autor, as grandes funções, interfaces e informações deveriam ser entendidas, primeiramente, antes das especificações das sucessivas camadas de detalhes.

De acordo com Sommerville [2007] e Tonsig [2008], os requisitos podem ser definidos como: a) funcionais, representando as funções do sistema e o que este deverá fazer; e b) não-funcionais, representando os atributos do sistema e as qualidades gerais do mesmo (segurança, manutenção, disponibilidade, desempenho entre outros). Os autores sugerem que os principais artefatos do fluxo de requisitos deverão conter os seguintes elementos: descrição do problema, glossário, atributos dos requisitos, matriz de rastreabilidade, especificações dos casos de uso e especificações suplementares (requisitos não- funcionais).

Relativamente às atividades de levantamento de requisitos, o grupo ISRD [2007, p.149] sugere que tais sejam considerados à luz de três objetivos principais: “antecipação, investigação e especificação de requisitos”. O primeiro diz respeito àqueles requisitos que são antecipáveis pela equipe de desenvolvimento em função de seu *know-how* e experiência; o segundo é o cerne da análise de requisitos, pois cuida de utilizar ferramentas e técnicas para explorar e documentar os requisitos; o terceiro, por fim, diz respeito à especificação dos requisitos, ou seja, a descrição das características desejadas no novo sistema.

A Análise de Requisitos, por sua vez, busca “ultrapassar barreiras de comunicação entre os clientes, os usuários e os analistas, para que os requisitos possam ser capturados e modelados corretamente” [Tonsig 2008, p.129]. De acordo com o autor, alguns aspectos são considerados essenciais no processo de desenvolvimento de *software*, tais como: conhecimento preciso do ambiente do utilizador, sua forma de trabalhar, a estrutura organizacional e, os problemas e as necessidades que o utilizador espera que sejam supridas pelo novo sistema. Ainda para Tonsig [2008, p.129], podem ser utilizados vários recursos para coletar os requisitos e entre estes encontram-se: “aplicação de questionários, inspeção de documentos utilizados nos processos, *brainstorm*, *Joint Application Design* (JAD), entrevistas, observação *in loco*, entre outros”. O entendimento de Tonsig [2008] está em plena concordância com o de Sommerville [2007], para o qual a Análise de Requisitos refere-se ao processo inicial do desenvolvimento de sistemas, que envolve a descoberta dos requisitos, bem como a análise e documentação dos mesmos.

Todavia, para se realizar um processo de análise de sistemas, não é suficiente sabermos o que são requisitos, levantá-los e analisá-los. Também se faz importante, para o sucesso do desenvolvimento do sistema, que os requisitos sejam devidamente gerenciados a fim de que estes forneçam as premissas para que o sistema seja desenvolvido de forma que auxilie na solução de problemas inerentes ao negócio, bem como atenda às necessidades ou restrições da organização [Tonsig, 2008; Chiu, 2005].

Ao longo do tempo, várias técnicas e métodos foram desenvolvidos com o intuito de auxiliar neste sentido e, assim, atender melhor as necessidades dos utilizadores, pois, conforme relata Parker [2000], na maioria dos sistemas efetivamente entregues o que se constata é que os mesmos não atendem à tais necessidades. Porém, observa-se que o levantamento e a caracterização dos requisitos ainda apresentam diversos problemas processuais. A importância de uma correta definição dos requisitos do sistema é ressaltada por vários autores [Jones 1994; Silva e Videira 2001; Chiu 2005 e Cerpa e Verner 2009]. Os mesmos afirmam que, quando mal elucidados, este fator torna-se o principal risco que pode levar ao fracasso da grande maioria dos projetos de desenvolvimento de sistemas.

Esta percepção, por sua vez, encontra eco nos resultados de uma pesquisa realizada pelo *Business Benchmark Report*, emitida em 2008 pelo *IAG Consulting*, de *New Castle*, onde a mesma relata que quando as empresas tinham uma definição requisitos pobre e uma fraca capacitação em análise de negócios, três quartos dos projetos fracassaram [Robb, 2009]. Da mesma forma, Eveleens e Verhoef [2010] citam um relatório do *Standish Group* em que se afirma que, em 2009, por exemplo, 68% dos projetos de desenvolvimento de sistemas ou falharam por completo, ou terminaram fora do prazo, fora do

orçamento ou com atendimento parcial dos requisitos definidos. A Figura 1, apresenta os resultados obtidos pelo relatório citado.

Standish project benchmarks over the years

| Year | Successful (%) | Challenged (%) | Failed (%) |
|------|----------------|----------------|------------|
| 1994 | 16 | 53 | 31 |
| 1996 | 27 | 33 | 40 |
| 1998 | 26 | 46 | 28 |
| 2000 | 28 | 49 | 23 |
| 2004 | 29 | 53 | 18 |
| 2006 | 35 | 46 | 19 |
| 2009 | 32 | 44 | 24 |

Figura 1 - Resultados dos estudos do *Standish Group*
Eveleens e Verhoef [2010, p.31]

Tais estatísticas são preocupantes, pois indicam a necessidade urgente de se buscar uma abordagem mais consistente para a definição de requisitos, pois que os erros encontrados nesta etapa devem ser sanados de imediato, a fim de se evitar que se propaguem para as etapas seguintes [Mecenas e Oliveira, 2005; Kotonya e Sommerville, 1998; Sommerville, 2007]. De acordo com este contexto, os autores Dinsmore e Cavelleri [2008] e Miguel [2009], evidenciam a necessidade de se gerenciar os requisitos, pois para os autores estes são a base para se estimar tamanho, esforço, cronograma e custo de um projeto. Portanto, ainda de acordo com os mesmos autores, faz-se necessário que as informações ali contidas sejam produzidas de forma que não venham a prejudicar o bom andamento do projeto durante a sua execução.

Diante de tantas preocupações e das inúmeras necessidades que aparecem durante o ciclo de vida do *software*, especialmente de manter um controle adequado de todas as etapas do projeto, as empresas buscam apoio em metodologias que visam a garantia da qualidade e o aumento da produtividade. De acordo com Booch, Rumbaugh e Jacobson [1999], a utilização de metodologias no desenvolvimento de sistemas possibilita: (1) Visualizar o sistema no tempo (presente, passado e futuro); (2) Especificar a estrutura, a arquitetura e o comportamento do sistema; (3) Conduzir e controlar o processo de construção do sistema; (4) Documentar as decisões tomadas. Note-se que tais autores foram os principais precursores da abordagem conceitual mais utilizada atualmente em desenvolvimento de sistemas, qual seja, a de orientação a objetos, a qual está no centro de metodologias importantes como a *Rational Unified Process* – IBM RUP.um outro contexto, de acordo com Miguel [2009, p.27], é bem conhecida a

“maturidade da área de Gestão de Projetos”, particularmente no que diz respeito às técnicas de levantamento e definição de âmbito, bem como de suas atividades internas. Segundo o referido autor [2009, p.1] esta área, anteriormente vista como um sistema burocrático interno às organizações, passou a ser “considerada como uma arma competitiva” que proporciona níveis mais elevados de eficiência, qualidade e valor acrescentado aos clientes.

Em realidade, muito da maturidade desta área de conhecimento decorreu do uso intensivo do guia PMBoK - *Project Management Body of Knowledge*, editado pelo *Project Management Institute* (PMI). Neste documento, apresenta-se uma visão geral de cada subconjunto do conjunto de conhecimentos da gestão de projetos, o que acaba por estabelecer uma linguagem padrão, comum a todos os profissionais da área de projetos, oferecendo um apoio inestimável ao seu gerenciamento, desde a fase de iniciação até o encerramento [PMBoK 2008, p.4]. Ainda de acordo com o guia PMBoK [2008, p.12], a área de Gestão de Projetos é vista como “a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos”. Atualmente, o guia PMBoK está em sua 4ª Edição, lançada no ano de 2008.

De todo modo, o PMBoK/PMI identifica três restrições fundamentais em um projeto qualquer, a saber – âmbito, prazo e custo. Estas compõem o chamado “triângulo de restrições do projeto” [Miguel 2009, p. 4], conforme ilustrado na Figura 2. A área interior do triângulo representa os recursos disponíveis e a qualidade almejada para o projeto e estes, por sua vez, estão limitados pelas linhas que representam o âmbito, prazo e custo.

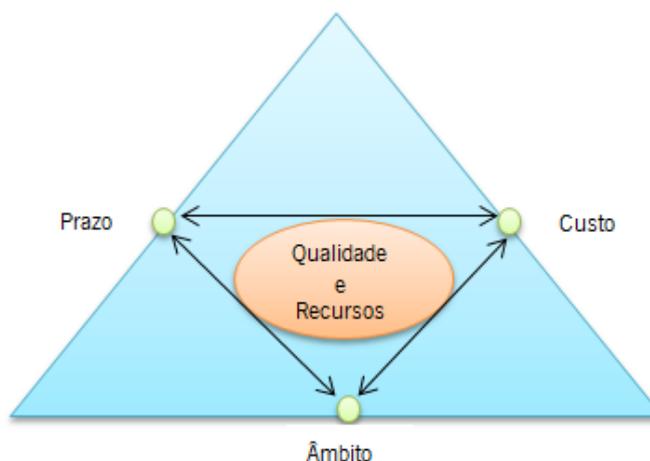


Figura 2 - Triângulo de restrições do projeto
Miguel [2009, p. 13]

De acordo com aquele mesmo autor, entende-se por *âmbito* como sendo o conjunto de requisitos para atingir os objetivos definidos; por *prazo*, o tempo pelo qual o projeto deve ser finalizado; e por *custo*, o dinheiro disponível para realizar o projeto.

De especial interesse para os fins desse estudo, a definição de âmbito em projetos, ainda segundo o PMBoK/PMI, está fundamentada em uma série de documentos específicos produzidos logo nas primeiras fases de um projeto, como a Declaração de Âmbito, o *Work Breakdown Structure (WBS)* e a Matriz de Responsabilidades, entre outros. Desta forma, tudo o que estiver incluído e definido em tais documentos constitui-se no que deve ser feito, inexoravelmente. Esta abordagem, simples e objetiva, usualmente estabelecida na forma de um contrato formal, traz robustez, consistência e rastreabilidade entre o que deve ser feito (âmbito) e o que deve ser entregue (produto/serviço).

Como se observa, então, os conceitos de requisitos (sistemas) e âmbito (projetos) guardam uma similaridade notável, pelo que pode-se inferir que algumas das técnicas utilizadas em gestão de projetos para definição de âmbito sejam aplicáveis, ainda que com pequenas adaptações, à descoberta e estruturação de requisitos funcionais em desenvolvimento de sistemas.

1.2 Objetivos do estudo

Tendo-se em conta as considerações apresentadas no tópico anterior e o fato de que, atualmente, as principais abordagens de desenvolvimento de sistemas atuais são orientadas a objetos, estabeleceu-se a seguinte questão de investigação: *É possível o uso de técnicas de definição de âmbito, da área de Gestão de Projetos, para uma melhor definição de requisitos em sistemas orientados a objetos?*

Para responder a essa questão, o presente trabalho será realizado em etapas, as quais corresponderão a objetivos a serem alcançados, conforme segue:

- a) Analisar as similaridades conceituais entre as áreas de Gestão de Projetos e de Análise Orientada a Objetos, a partir de seus respectivos documentos de definição de âmbito/requisitos;
- b) Propor um processo de transposição de técnicas de definição de âmbito, em Gestão de Projetos, devidamente adaptados ao contexto de análise orientada a objetos, o qual permita uma elaboração mais abrangente de artefatos em levantamento de requisitos; e
- c) Aplicar o processo proposto em um sistema orientado a objetos existente, afim de se experimentar a sua viabilidade prática.

Ao final desta dissertação, pretende-se que a abordagem proposta auxilie no levantamento/extração de requisitos de forma mais completa e coesa, recorrendo à utilização concertada de técnicas de gestão de projetos e de delineamento de requisitos de sistemas orientados a objetos.

1.3 Abordagem metodológica

De acordo com Berndtsson, Hansson, Olsson e Lundell [2008, p.12], um método representa o “meio, procedimento ou técnica utilizada para realizar um processo de uma forma lógica, ordenada e sistemática”. Quando se trata de um projeto de pesquisa, um método refere-se a uma “abordagem organizada para a resolução de problemas que inclui: (1) a coleta de dados, (2) a formulação de uma hipótese ou proposição, (3) o teste da hipótese, (4) a interpretação dos resultados, e (5) alcançar conclusões que posteriormente possam ser avaliadas de forma independente por outros”. De acordo com os autores, um dos principais objetivos ao se desenvolver uma dissertação é “obter capacitação na utilização de um método científico, o qual pode ser aplicado durante a estruturação e resolução de problemas mais complexos”.

Ainda para aqueles autores, existem outros tipos de métodos que podem ser utilizados nas áreas de Ciência da Computação e Sistemas de Informação, como por exemplo, Método Quantitativo e Método Qualitativo. Relativamente a esta dissertação, a abordagem seguida foi a de *Design Science Research*. Segundo Aken [2005], a abordagem *Design Science* refere-se à agregação de um conhecimento novo ao *design* de soluções, normalmente relativas a problemas existentes no mundo real. Para além de agregar ou gerar conhecimento, ainda segundo aquele autor, tal abordagem deverá fornecer ferramentas adequadas para que os profissionais, de diversas áreas, possam utilizá-las da melhor forma. Wastell, Sauer e Schmeink [2009], em consonância com Aken, afirmam que o objetivo principal desta abordagem é a incorporação de conhecimentos orientados pelas práticas de implementação, gerenciamento e uso de artefatos.

Manson [2006], por sua vez, entende que o processo de utilizar o conhecimento para o planeamento e criação de um artefato, quando realizado de forma cautelosa e analisado com rigor no que diz respeito à efetividade com que se atingem os objetivos propostos, pode ser nomeado de pesquisa e, a esta sua forma, de *Design Research*.

Diante deste contexto, o que se pretende com a utilização da abordagem proposta nesta dissertação é prestar um contributo na área de pesquisa em questão. Objetivamente, pretende-se a produção ou adequação de artefatos de gerenciamento de projetos, para que estes possam auxiliar os profissionais de

desenvolvimento de sistemas de informação, e, assim, venham a ter um maior nível de qualidade nos documentos produzidos nas fases iniciais do desenvolvimento de sistemas. Para tanto, é necessário proceder a um estudo detalhado dos artefatos típicos para a definição de âmbito no contexto de gerenciamento de projetos, bem como verificar a possibilidade de estes serem utilizados no desenvolvimento de sistemas, particularmente nas fases de levantamento e análise de requisitos em sistemas orientados a objetos.

A fim de obter um embasamento teórico ao estudo pretendido, foi realizada uma ampla pesquisa sobre as principais características das metodologias de desenvolvimento de sistemas (estruturada e orientada a objetos) e de técnicas de gestão de projetos, descritas nos Capítulos 2 e 3, respectivamente. Desta forma, no que diz respeito a metodologias de desenvolvimento de sistemas, a referida pesquisa foi centrada no *Rational Unified Process* - IBM RUP. Por outro lado, quanto aos conceitos de Gestão de Projetos, o enfoque foi a metodologia preconizada pelo *Project Management Institute* - PMI, veiculada por meio de seu *Project Management Body of Knowledge* (PMBok).

Para além das referências bibliográficas citadas ao final deste relatório, as quais foram utilizadas para a referido embasamento teórico, também foram buscadas informações adicionais em serviços e publicações de referência, como: *Web of Knowledge*, *Scopus*, *NDLTD*, *Driver*, *Google Scholar* e Repositório da UMinho.

Por fim, por meio de estudo de caso com base em um projeto desenvolvido, nomeadamente o *Sistema de Apoio à Qualidade da Santa Casa de Misericórdia de Guimarães - SIAQ-SCMG*, foi experimentada a aplicabilidade dos artefatos produzidos (ou adequados) com as subseqüentes considerações sobre os resultados obtidos.

Assim, a Figura 3, abaixo, sintetiza as fases da abordagem metodológica seguida, conforme descrita, as quais permitiram o alcance dos objetivos traçados no tópico 1.2.

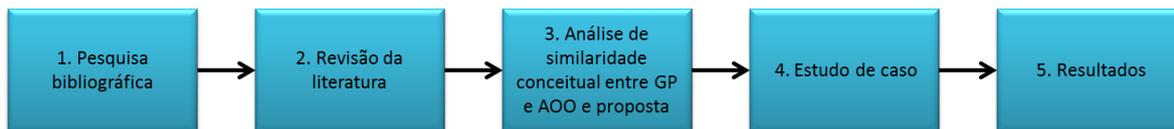


Figura 3 - Fases da abordagem metodológica em curso

1.4 Estrutura do documento

Relativamente à estrutura deste documento, a presente dissertação desenvolver-se-á no decorrer de 5 capítulos, para além deste. No capítulo 2, discorre-se sobre os conceitos relevantes em Análise de Sistemas, apresentando-se um breve histórico das Análises Estruturada e Essencial. Em seguida, o mesmo apresenta maiores detalhes sobre a Análise Orientada a Objetos, com ênfase em seus principais artefatos. Ainda naquele capítulo, aborda-se a metodologia *Rational Unified Process - RUP*, com enfoque principal nas suas disciplinas Modelagem de Negócios, Requisitos e Análise e Design, bem como nas fases de Iniciação e Elaboração.

O Capítulo 3, por sua vez, contém um breve relato histórico da área de Gestão de Projetos, mas, para além disso, seu principal objetivo é apresentar conceitos relevantes acerca do ciclo de vida de um projeto, bem como as principais técnicas e documentos relacionados ao delineamento do âmbito, com destaque para o Termo de Abertura de Projeto, Declaração de Âmbito, *Work Breakdown Structure - WBS*, Matriz de Responsabilidades e Diagrama de Precedência, as quais serão as peças fundamentais para a proposta de conexão entre as áreas de Gestão de Projetos e Análise Orientada a Objetos.

Desta forma, o Capítulo 4 apresenta a análise de similaridade conceitual entre aquelas áreas, a qual trata-se de um dos objetivos do presente trabalho. Assim, por meio dos principais documentos/artefatos utilizados para delineamento de âmbito/requisitos em ambas as áreas, procura-se buscar pontos de semelhança entre eles, a fim de se estabelecer uma estratégia de conexão entre os mesmos. Após cumprida esta tarefa, foi possível estabelecerem-se passos práticos que permitem viabilizar tal proposta, da qual algumas vantagens são esperadas.

O Capítulo 5, por seu turno, procura colocar em prática a proposta de conexão, ou seja, experimenta-se a técnica sugerida no capítulo anterior a um sistema real, a fim de se verificar a sua aplicabilidade no contexto de desenvolvimento de sistemas orientados à objetos. Desta forma, apresenta-se brevemente a instituição escolhida para tal estudo de caso, bem como alguns aspectos gerais sobre a documentação do sistema obtida. Assim, a partir de tal sistema e sua documentação, aplica-se o processo proposto e, por fim, expõem-se algumas considerações acerca dos resultados obtidos.

Por fim, o Capítulo 6 apresenta algumas conclusões acerca do trabalho realizado, bem como algumas considerações relativas as suas limitações e possibilidades de ampliação em trabalhos futuros.

2. Conceitos relevantes em Análise de Sistemas

2.1 Introdução

Em que pese a área de desenvolvimento de sistemas, atualmente, estar fortemente apoiada no conceito de objetos, seja ao nível da análise, seja ao nível da programação, ou, ainda, ao nível da base de dados, a verdade é que tal conceito deve muito às abordagens que lhe antecederam, nomeadamente, à Análise Estruturada e à Análise Essencial.

Apesar do sucesso indiscutível da abordagem orientada a objetos, nota-se a proliferação de artefatos utilizados na modelagem de um sistema informatizado, como no caso da IBM RUP, onde existem inúmeros artefatos previstos. Naturalmente, tal proliferação decorre do aumento da complexidade dos sistemas informáticos modernos. Todavia, também por conta deste fato, parece evidente um aumento da dificuldade de comunicação entre clientes e desenvolvedores de sistemas. A consequência desta realidade está refletida na indiscutível dificuldade de se estabelecer, de forma correta e completa, o conjunto de requisitos que atendem às necessidades do cliente, conforme longamente discutido no capítulo 1 deste estudo.

Outra consequência do elevado número de artefatos atualmente previstos nas abordagens de Análise Orientada a Objetos, é a possível perda da visão global do sistema e, por conseguinte, dos requisitos que tratam da integração entre os demais requisitos. Em outras palavras, perde-se a visão integradora dos diversos módulos funcionais (ou de negócio) do sistema, problema este não atacado pelas ditas abordagens. Por outro, as abordagens de desenvolvimento iniciais, como as já citadas Estruturada e Essencial, tinham grande preocupação com este importante aspecto.

Assim, tendo-se em conta que o presente estudo pretende utilizar algumas técnicas de Gestão de Projetos em Análise Orientada a Objetos, particularmente no que diz respeito à definição de requisitos, torna-se importante compreender a importância e as formas de se obter aquela visão global, além, obviamente, da visão mais fragmentada proporcionada pela abordagem orientada a objetos, que também será alvo de considerações específicas.

2.2 Documentos em Análise Estruturada e Essencial

Segundo Pressman [1997, p. 315], os primeiros esforços para a elaboração de uma abordagem metodológica ocorreram no final dos anos 1960 e começo dos anos 1970. De acordo com o mesmo autor [p. 351], a Análise Estruturada surgiria apenas em 1977 com o trabalho de Ross e Schoman acerca da *Structured Analysis and Design Technique - SADT*, sendo esta, porém, popularizada por Tom DeMarco. De fato, o próprio DeMarco reconhece o pioneirismo de Ross ao afirmar que “a primeira vez que viu o adjetivo ‘estruturado’ aplicado à diagramas em níveis”, foi num trabalho de Doug Ross, ainda em 1974 [DeMarco 2001, p. 523]. De todo o modo, as contribuições de Ross e DeMarco ajudaram os analistas de sistemas a criar modelos de fluxo de informação a partir de símbolos gráficos, dicionários de dados e narrativas de processos, entre outras técnicas [Pressman 1997, p. 315].

Deste modo, de acordo com Tonsig [2008, p.145], a Análise Estruturada tinha como principal propósito facilitar a “comunicabilidade e clareza dos objetivos do sistema” e, ainda segundo o próprio autor, o conceito fundamental utilizado por ela era o de “construção de um *modelo lógico* (não físico) de um sistema”. Essa metodologia propõe etapas para o desenvolvimento de um sistema, conforme ilustrado pela Figura 4.

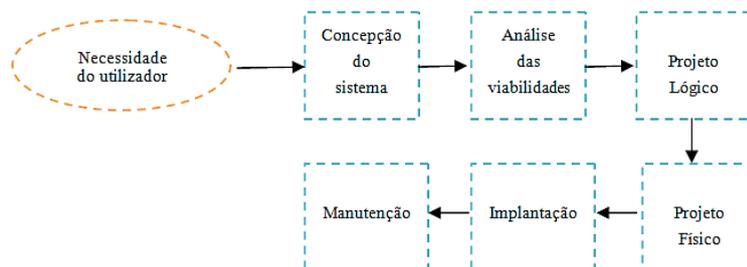


Figura 4 - Desenvolvimento de Sistema pelo Método da Análise Estruturada
Tonsig [2008, p.146]

Assim, desenvolver um sistema com método de análise estruturada significa construí-lo de forma *top-down*, onde deverão ser realizados refinamentos sucessivos para a produção de uma visão global do sistema, sendo esta alcançada através do emprego do Diagrama de Fluxo de Dados - DFD [Tonsig 2008, p.146]. A Figura 5, a seguir, exemplifica um requisito de sistema definido como “Cadastrar pedido”, o qual relaciona-se com os seguintes depósitos de dados: Cliente, Transportadora, Pedido, Estoque e Produto.

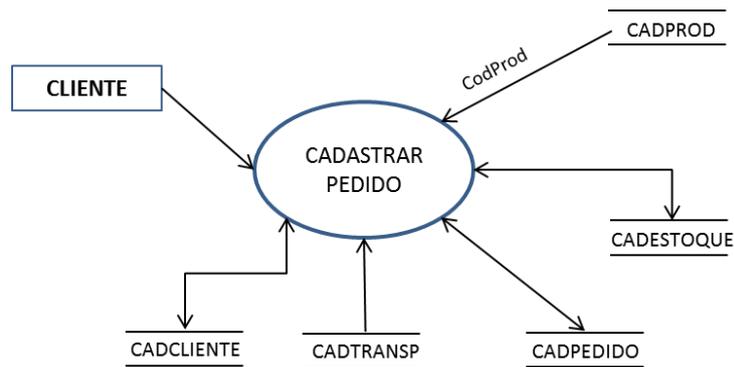


Figura 5 - Exemplo de DFD típico em Análise Estruturada
Tonsig [2008, p.155]

Conforme se pode observar, este tipo de diagrama contém a definição do fluxo de dados entre os vários processos funcionais, os quais podem ser entendidos como a “transformação de dados de entrada em dados de saída de acordo com algumas funções” [Finkelstein, Huang, Finkelstein e Nuseibeh 1992, p.5].

Com base neste DFD, ainda de acordo com estes últimos autores, é feita uma decomposição funcional, onde criam-se outros fluxos menores, os quais são um detalhamento do fluxo macro. Ao proceder-se deste modo, obter-se-ão os contornos dos dados requeridos, que posteriormente, serão o ponto de partida para a elaboração de um Diagrama de Entidades Relacionamentos – DER, o qual, por sua vez, possui a finalidade de “mostrar as entidades de dados e como estão relacionadas entre si” [Finkelstein, Huang, Finkelstein e Nuseibeh 1992, p.5].

Apesar das importantes contribuições da Análise Estruturada, Satzinger [1993] relata que, em meados dos anos de 1980, existia uma percepção dúbia quanto àquela abordagem. Basicamente, havia questionamentos sobre o tempo gasto entre a modelagem física e sua transposição para o modelo lógico com os novos requisitos, bem como sobre o uso adequado do DFD. Uma das consequências desse contexto, segundo ainda esse autor, foi o surgimento da *Análise Essencial de Sistemas*, introduzida por McMenamin e Palmer, a qual procurava dar um tratamento adequado àquelas questões citadas.

A metodologia proposta pela Análise Essencial, segundo Tonsig [2008, p.170], pode ser considerada um refinamento da Análise Estruturada. De acordo com o autor, o objetivo básico da Análise Essencial é “descrever o sistema de maneira independente de restrições tecnológicas”. Assim, para esta metodologia, um problema existente deverá ser estudado, porém não deverá ser modelado de imediato. De fato, os esforços primeiros deverão ser concentrados na identificação das funcionalidades lógicas

requeridas para o sistema e, somente após esta etapa, produzir apenas um modelo lógico do sistema - o *modelo essencial* – o qual não conterà, por conseguinte, as exigências físicas e/ou tecnológicas.

Na Análise Essencial, o sistema é visto como um conjunto de respostas planejadas por meio de atividades que reagem a estímulos provocados por eventos no ambiente. Estes eventos, por sua vez, podem ser de caráter externo (fora do ambiente de controle do sistema) ou temporal (como consequência do decorrer do tempo) [Satzinger, 1993]. O seu processo é composto por dois modelos, Ambiental e Comportamental, conforme ilustrado pela Figura 6.

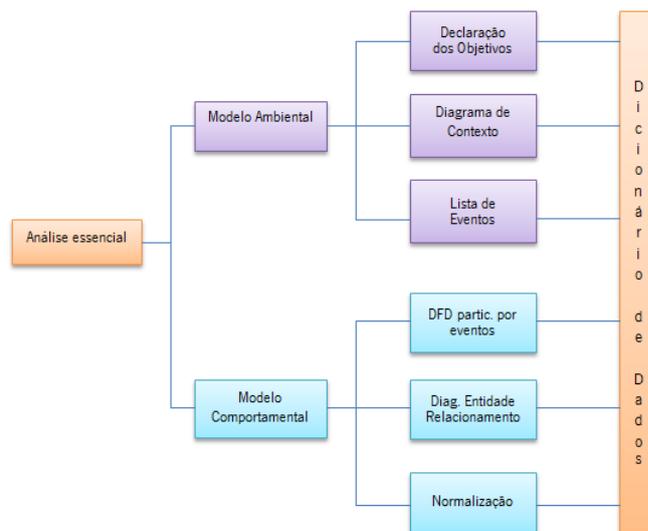


Figura 6 - O Modelo Essencial
Tonsig [2008, p.171]

Conforme se pode observar pela figura anterior, para além da questão da divisão em dois modelos, há, basicamente, duas novidades na modelagem em relação à Análise Estruturada: “Lista de Eventos” e “DFD Particionado por Eventos”. Ainda conforme Satzinger [1993], o objetivo da Lista de Eventos, elaborada junto ao cliente no Modelo Ambiental, era funcionar como base para o particionamento do sistema em módulos fracamente acoplados. Numa etapa seguinte, já no Modelo Comportamental, seriam elaborados DFD para cada um dos eventos listados, compondo-se, então, o chamado “DFD Particionado por Eventos”. Após isto, estes DFD Particionados deveriam ser consolidados, compondo, finalmente, um DFD de nível 0 (zero).

Observa-se, assim, que, enquanto a Análise Estruturada têm um carácter *top down*, a Análise Essencial possui uma característica *bottom-up*, a qual permite ao analista e ao cliente uma visão global do sistema, pois facilita uma melhor percepção de suas partes constituintes, bem como da integração entre estas.

A Tabela 1, a seguir, apresenta um resumo das abordagens citadas, bem como suas principais ferramentas no levantamento e caracterização de requisitos.

| Modelo e principais metodologistas | Ênfase | Ferramentas |
|--|--|--|
| Estruturado Chris Gane Edward Yourdon Trish Sarson Tom de Marco | Funcionalidade e dados | Diagrama de Fluxo de Dados (DFD), Especificação de processos, Diagrama de Entidade Relacionamento (DER), Normalização, Diagrama de Estrutura de dados (DED) e Dicionário de dados. |
| Essencial Sthepehn McMenamim John Palmer | Essência funcional e dados Integração funcional e dados | DFD de Contexto, Lista de Eventos, DFD particionado por eventos, Diagrama Entidade Relacionamentos (DER), Diagrama de Estrutura de Dados (DED), Normalização, Dicionário de Dados. |

Tabela 1 - Resumo das abordagens metodológicas
Tonsig [2008, p.144]

Também pode-se observar que estas duas abordagens tratam separadamente os aspectos funcional e de dados, o que torna difícil integrar essas duas visões típicas em desenvolvimento de sistemas. Não por acaso, a nova metodologia, que viria a seguir, fundiria esses dois aspectos num novo conceito, nomeadamente, o de objeto, o qual deu origem à Análise Orientada a Objetos.

2.3 Artefatos em Análise Orientada a Objetos

Segundo Pressman [1997, p.567] a abordagem orientada a objetos foi inicialmente proposta no final dos anos 60. Entretanto, foram necessários 20 anos para que se tornasse amplamente utilizada, de forma que, ainda segundo o mesmo autor, na primeira metade dos anos 1990, a engenharia de sistemas orientados a objetos havia se tornado o paradigma escolhido pelo mercado.

Buscando eliminar a falta de padronização do paradigma de orientação a objetos, James Rumbaugh (*Object Modeling Technique* - OMT) e Grady Booch (*Booch Method*) trabalharam em conjunto, em 1994, com o objetivo de unificar suas abordagens. Em 1995, Ivar Jacobson com o método OOSE (*Object - Oriented Software Engineering*), uniu-se ao grupo e em 1997 formalizaram a *Unified Modeling Language - UML*, que veio a ser adotada como padrão pelo *Object Management Group - OMG* [Dobing e Parsons, 2008].

De acordo com Jacobson, Booch e Rumbaugh [1999, p.3], a UML “é uma linguagem de modelagem” de propósito geral utilizada para “especificar, visualizar, construir e documentar os artefatos de um sistema de *software*”. É uma linguagem de modelagem que visa “unificar a experiência passada sobre técnicas de modelagem e incorporar as atuais melhores práticas em uma abordagem padrão”. Neste sentido, esta abordagem trata o sistema como sendo constituído de partes “estática, dinâmica, ambiental e organizacional”, suportando a maioria dos processos de desenvolvimento orientados a objetos.

A UML “é vista como a notação padrão, de fato e de direito, para a modelagem de sistemas orientados a objetos”, auxiliando nos processos de análise, projeto e construção de *softwares* [Larman 2008, p.43]. Para tanto, disponibiliza diversas notações para a caracterização de um sistema sob diferentes perspectivas, o que permite visões parciais dos elementos que o compõem. Desta forma, os principais benefícios que a linguagem oferece são três: visualização, gestão de complexidade e comunicação clara [Rational, 2012]. Abaixo, apresentam-se os diagramas mais utilizados para modelagem de sistemas orientados a objetos [Jacobson, Booch e Rumbaugh 1999, p.26]:

- **Diagrama de casos de uso:** designa um conjunto de casos de uso e atores e seus relacionamentos. Abrangem a visão estática de casos de uso do sistema. São considerados muito importantes para a organização e modelagem de comportamento do sistema, pois “constituem a técnica em UML para representar o levantamento de requisitos de um sistema” [Nunes e O'Neill 2008, p. 57]. A Figura 7, a seguir, apresenta-nos um exemplo deste diagrama.

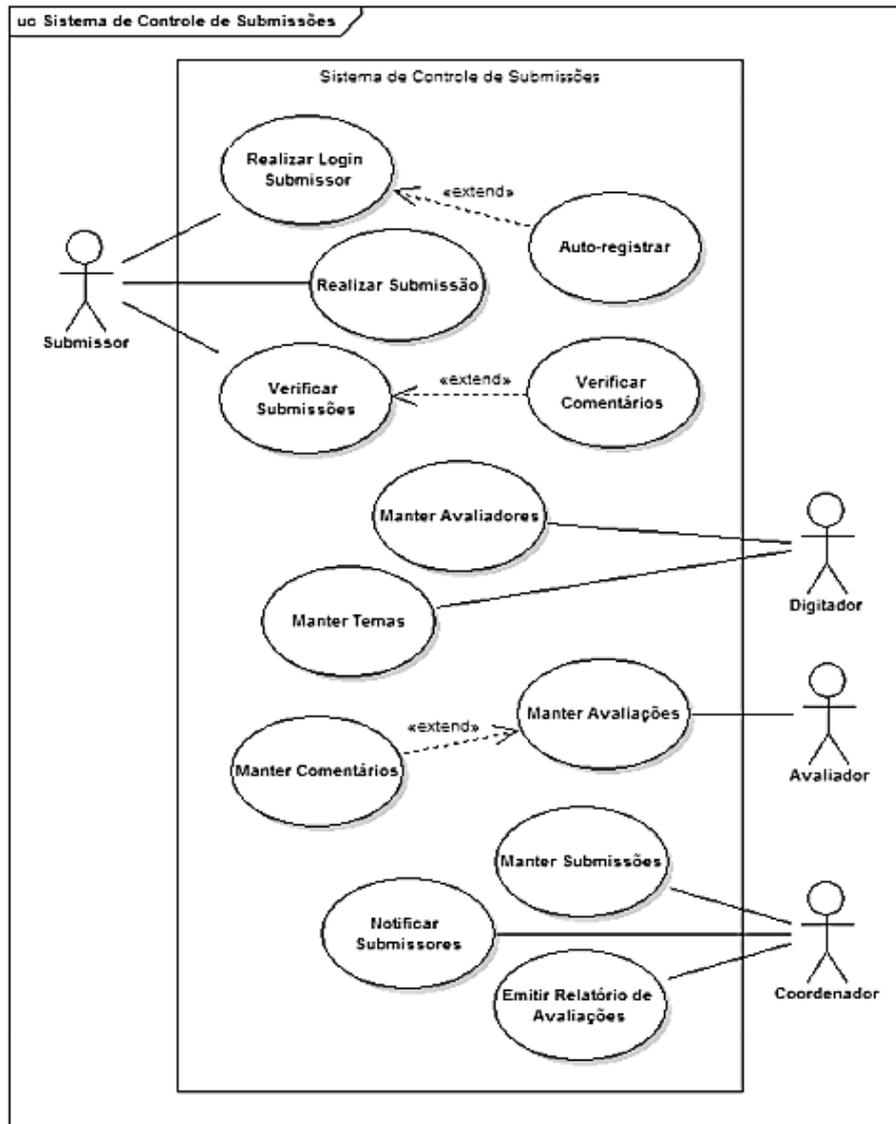


Figura 7 - Diagrama de Caso de Uso
Guedes [2005, p.8]

- **Diagrama de estados:** trata-se de uma máquina de estados, formada por estados, transições, eventos e atividades. Semelhantemente aos diagramas de sequência e colaboração, abrangem a visão dinâmica do sistema e, nesse sentido, conforme exemplo apresentado na Figura 8, é utilizado para “mostrar os possíveis estados dos objetos de uma classe no decorrer do seu ciclo de vida” [Tonsig 2008, p.280].

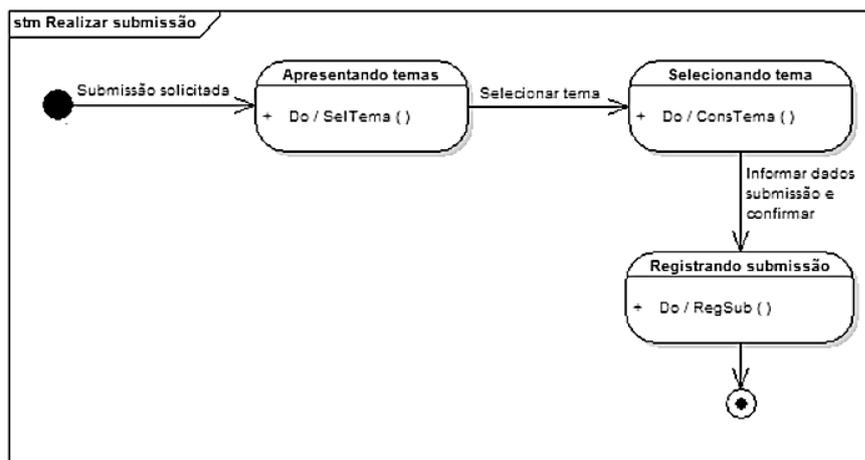


Figura 8 - Diagrama de Estados
Guedes [2005, p.11]

- **Diagrama de atividades:** expõe o fluxo de encadeamento de uma atividade para outra no sistema. Tem uma grande importância para a modelagem das funções de um sistema e dá ênfase ao fluxo de controle entre objetos. Neste sentido, deve-se entender uma atividade como sendo “uma etapa de um processo onde se realiza algum trabalho. A atividade pode ser um *caso de uso*, uma operação de uma classe, um grupo de casos de usos relacionados entre si” [Lopes, Morais, e Carvalho 2009, p. 136]. Na Figura 9, a seguir, tem-se um exemplo da aplicação deste tipo de diagrama.

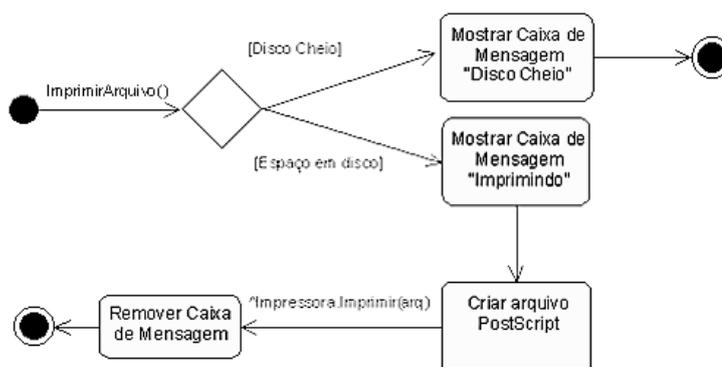


Figura 9 - Diagrama de atividades
Guedes [2005, p.12]

- **Diagrama de classes:** expõe um conjunto de classes, *interfaces* e colaborações, bem como os seus relacionamentos. A criação desse diagrama resulta de um “processo de abstração através do qual se identificam os objetos (entidades e conceitos) relevantes no contexto que se pretende modelar e se procuram descrever características comuns em termos de propriedades (atributos)

e de comportamento (operações)” [Nunes e O’Neill 2008, p.35]. A Figura 10 apresenta um exemplo prático da utilização deste tipo de diagrama.

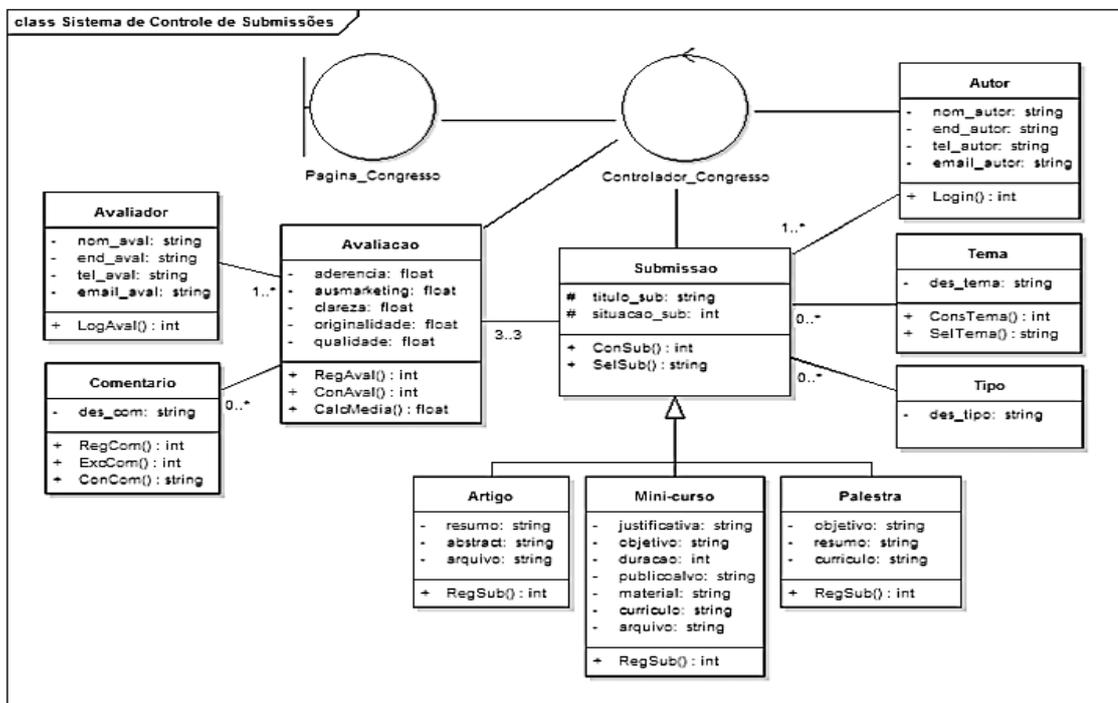


Figura 10 - Diagrama de Classes

Guedes [2005, p.8]

- **Diagrama de objetos:** contém um conjunto de objetos e seus relacionamentos. Representa “retratos” de instâncias de itens encontrados em diagramas de classes. Guedes [2005, p.46] reforça o entendimento de Jacobson, Booch e Rumbaugh [1999, p.26], ao afirmar que este diagrama tem como objetivo “fornecer uma ‘visão’ dos valores armazenados pelos objetos das classes definidas no Diagrama de Classes em um momento da execução de um determinado processo do sistema”. Nesse sentido, conforme se observa pelo exemplo da Figura 11, o autor faz notar que “os objetos não apresentam métodos, somente atributos e estes armazenam os valores possuídos pelos objetos em uma determinada situação” [Guedes 2005, p.46].

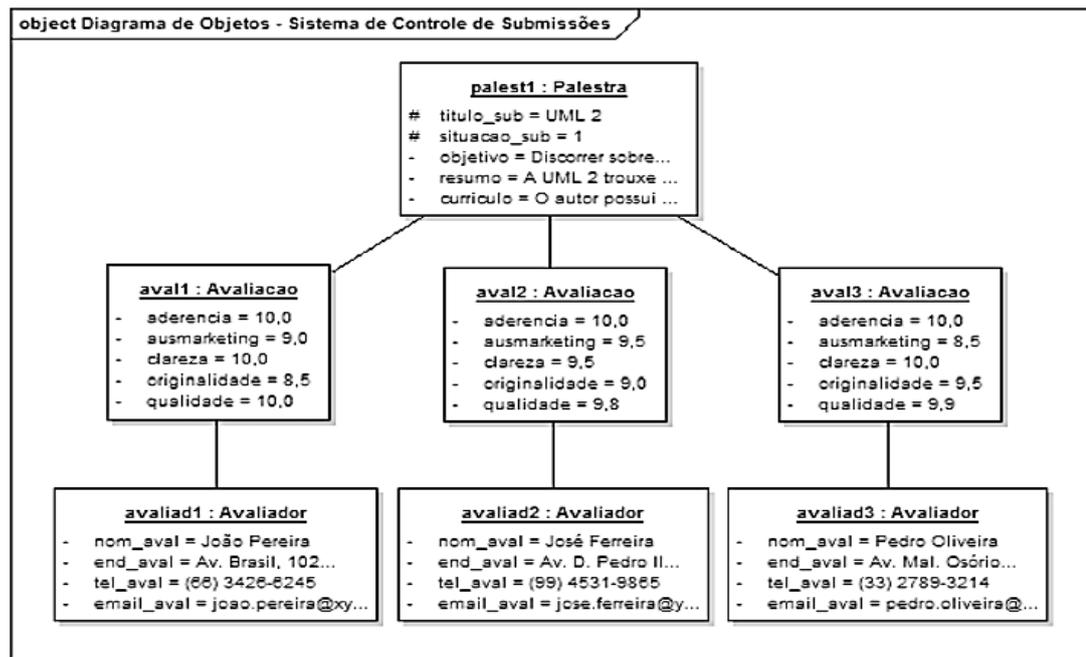


Figura 11 - Diagrama de Objetos
Guedes [2005, p.9]

- **Diagramas de sequência e colaboração:** trata-se de diagramas de interações, os quais consistem de um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles. Os diagramas de interações abrangem a visão dinâmica do sistema. Um diagrama de sequência é “um diagrama de interação cuja ênfase está na ordenação temporal das mensagens”. Quanto ao diagrama de colaborações, a ênfase está na “organização estrutural dos objetos que enviam e recebem mensagens”. O Diagrama de Colaboração passou a chamar-se Diagrama de Comunicação a partir da versão 2.0 da UML. [Guedes, 2005, p.10]. As Figura 12 e Figura 13, a seguir, trazem exemplos práticos dos referidos diagramas.

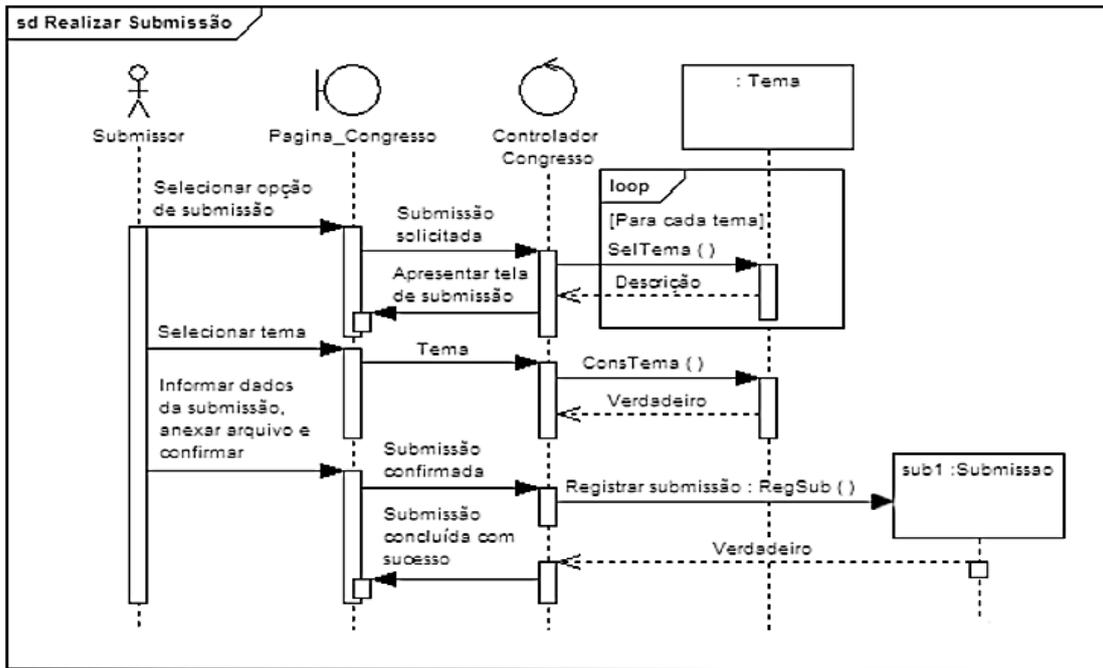


Figura 12 - Diagrama de Sequência
Guedes [2005, p.10]

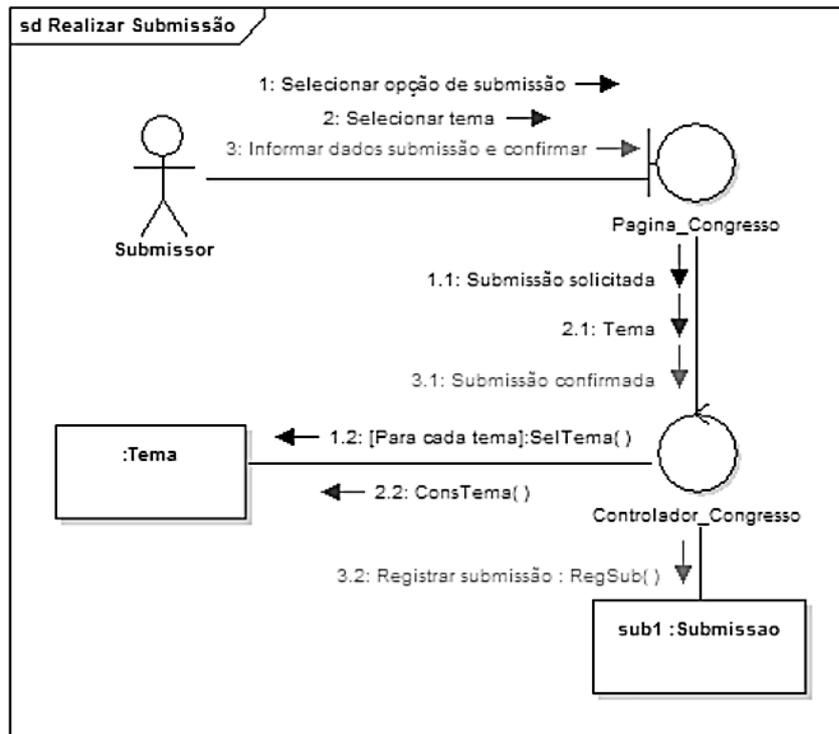


Figura 13 - Diagrama de Colaboração/Comunicação
Guedes [2005, p.11]

- **Diagrama de componentes:** contém as organizações e dependências existentes em um conjunto de componentes de *software*. Está relacionado aos diagramas de classe devido ao fato de que, geralmente, os componentes são mapeados para uma ou mais classes, *interfaces* ou colaborações. Segundo Lopes, Morais e Carvalho [2009, p. 141], os referidos *softwares* podem ser: programas fonte, ficheiros executáveis, bibliotecas e *scripts*, entre outros. Ainda segundo os mesmos autores, “um componente é, a nível físico, o equivalente de um pacote a nível lógico”. A Figura 14 exemplifica um Diagrama de Componentes referente a bibliotecas de programas utilizadas por uma aplicação (arquivo executável).

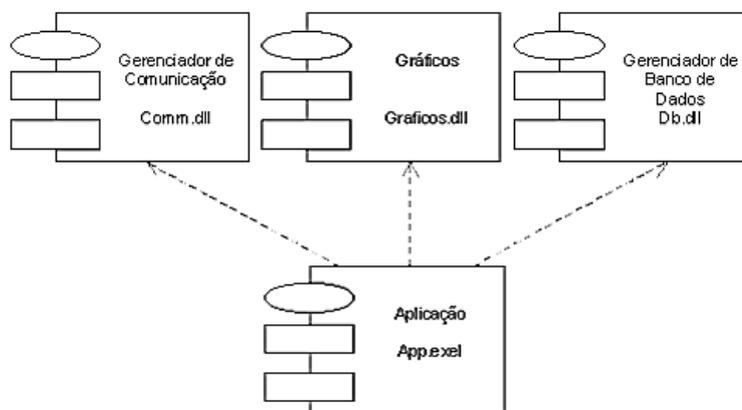


Figura 14 - Diagrama de Componentes
Guedes [2005, p.13]

- **Diagrama de implantação:** apresenta a configuração dos “nós” de processamento em tempo de execução e os componentes neles existentes. Lopes, Morais e Carvalho [2009, p. 143] lembram, ainda, que este diagrama “modela a arquitetura física de *hardware* mostrando as dependências e as *interfaces* entre os nós”, os quais representam “os recursos físicos onde são executadas as componentes e são computadores ou outros dispositivos (*modems*, impressoras,...)”. A Figura 15, abaixo, exemplifica sua utilização prática.

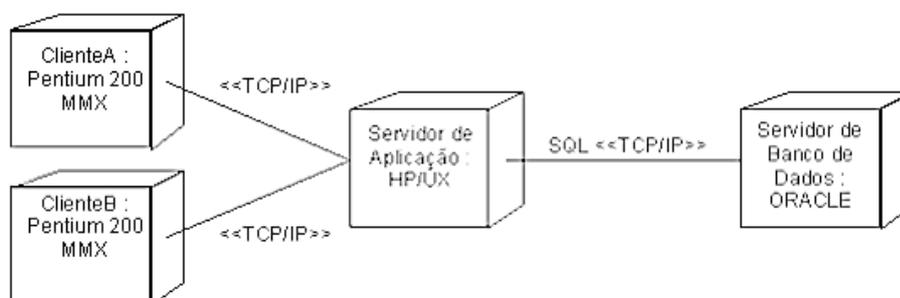


Figura 15 - Diagrama de Implantação
Guedes [2005, p.14]

Após vistos e compreendidos todos estes diagramas, nota-se que, apesar de sua flexibilidade e diversidade, o processo de desenvolvimento de um sistema orientado a objetos não está contido na UML, em que pese a mesma ser uma notação aplicável para a “análise de requisitos, análise sistêmica, projeto, implementação e testes” [Tonsig 2008, p.239]. Nesse sentido, a UML não é uma metodologia de desenvolvimento de sistemas orientados a objetos [Jacobson, Booch e Rumbaugh 1999, p.8].

Por outro lado, a metodologia orientada a objetos mais utilizada no mercado é a *IBM Rational Unified Process* (IBM RUP) [Larman 2008, p. 46; Gartner, 2011], a qual contém “um *framework* de processos abrangentes que fornecem práticas testadas na indústria para a implementação e entrega de aplicativos e sistemas e para um gerenciamento de projeto efetivo” [IBM, 2012]. Trata-se, efetivamente, de um “processo de engenharia de *software*” que “oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento” [Rational, 2006].

A metodologia IBM RUP divide o desenvolvimento do sistema em quatro fases: Iniciação, Elaboração, Construção e Transição, as quais cobrem todo o ciclo de vida do desenvolvimento do sistema. Estas, por sua vez, relacionam-se, em maior ou menor grau, com diversas *disciplinas*, também previstas no IBM RUP. A Figura 16, a seguir, apresenta este inter-relacionamento, na medida em que evidencia o grau de relevância de cada disciplina em cada fase.

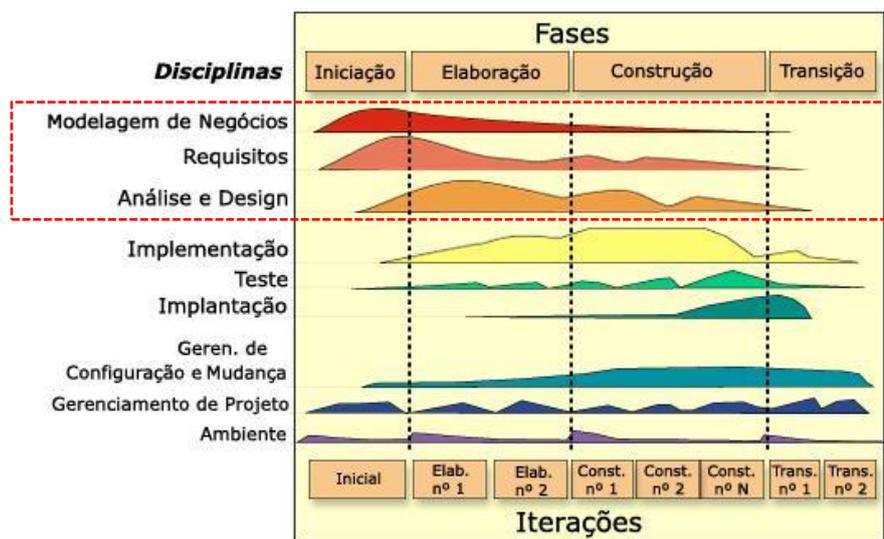


Figura 16 - Relacionamento entre fases e disciplinas segundo o IBM RUP Rational [2006]

Ao término de cada fase, os desenvolvedores deverão ter produzido um conjunto de artefatos previstos nessa metodologia, de forma que estes constituem-se nos marcos (*milestones*) finais de cada etapa. A Figura 17, a seguir, ilustra este aspecto.

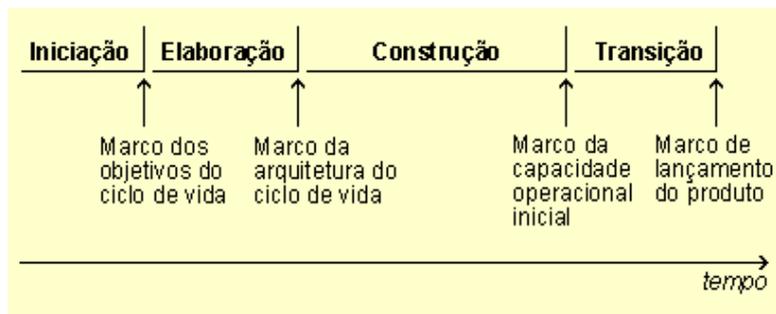


Figura 17 - As fases e os marcos de um projeto Rational [2006]

Conforme se pode observar na Figura 16, as fases de **Iniciação** e **Elaboração** constituem-se nos momentos de interesse deste estudo, pois é onde serão definidos os contornos do sistema, ou de outro modo, seus principais requisitos.

Neste sentido, ainda segundo a metodologia IBM RUP [Rational, 2006], o principal objetivo da fase de **Iniciação** é, então, obter um entendimento comum, entre os envolvidos, acerca dos objetivos do *software* do projeto. Deste modo, dois de seus principais objetivos se relacionam diretamente com o propósito deste estudo:

- “Estabelecer o **escopo [âmbito] do *software* do projeto e as condições limite**, incluindo uma visão operacional, critérios de aceitação e o que deve ou não estar no produto.
- Discriminar os **casos de uso críticos do sistema**, os principais cenários de operação e o que direcionará as principais trocas de *design*”. [Rational 2006 – grifos da autora]

Conforme apresentado na Figura 16, as principais disciplinas inerentes a essa fase são a) Modelagem de negócios; e b) Requisitos. Por conseguinte, no que diz respeito aos artefatos a serem produzidos e entregues, a Figura 18 apresenta uma síntese dos mesmos.

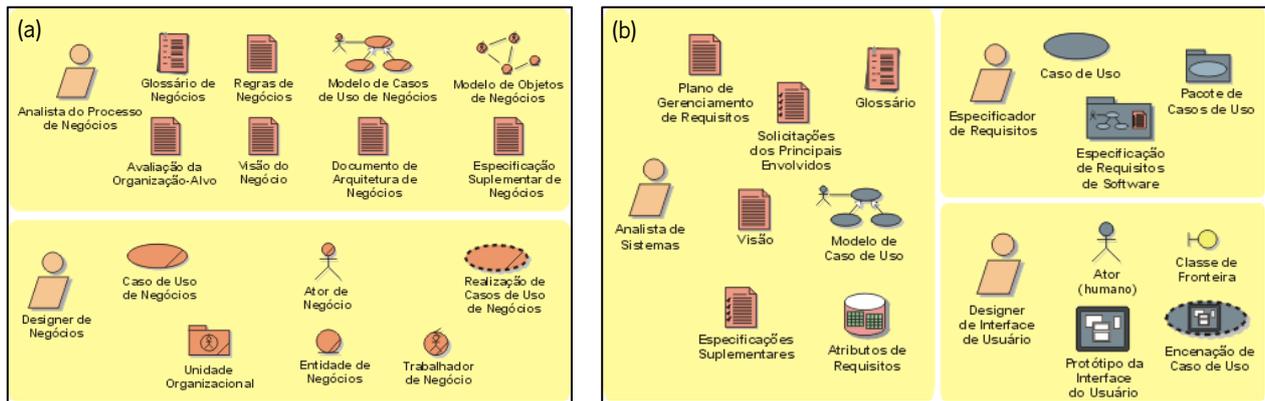


Figura 18 - Artefatos das disciplinas (a) *Modelagem de Negócios* e (b) *Requisitos*, respectivamente.

Rational [2006]

Conforme pode-se observar pela Figura 18, o Diagrama de Casos de Uso constitui-se, efetivamente, no principal artefato UML a ser produzido, pois, ao estar inserido tanto na disciplina Modelagem de Negócios quanto na disciplina Requisitos, propicia uma conexão entre as mesmas. Para além disso, partes específicas de um caso de uso também estão destacadas, como por exemplo, o caso de uso propriamente dito e os atores (humanos) do sistema. Vale observar, ainda, que os Casos de Uso podem ser especificados por meio de Diagramas de Atividade, conforme explicado anteriormente.

A fase de **Elaboração**, por sua vez, deverá propiciar uma estruturação conceitual que sustente as atividades que serão realizadas na fase de Construção. Neste sentido, “**a arquitetura se desenvolve a partir de um exame dos requisitos mais significativos** (aqueles que têm grande impacto na arquitetura do sistema) e de uma avaliação de risco” [Rational 2006 – grifos da autora]. Dentre os objetivos da fase de Elaboração, destacam-se os seguintes, por se relacionarem com o propósito deste trabalho:

- “Assegurar que a **arquitetura, os requisitos e os planos sejam estáveis**”;
- “Estabelecer uma arquitetura da **baseline derivada do tratamento dos cenários significativos** do ponto de vista da arquitetura” [Rational 2006 – grifos da autora].

Ainda de acordo com a Figura 16, observa-se que a principal disciplina inerente a essa fase é a de Análise e *Design*. Assim, no que diz respeito aos artefatos a serem produzidos e entregues quando de seu término, a Figura 19 apresenta uma síntese dos mesmos.



Figura 19 - Artefatos da disciplina "Análise e *Design*"
Rational [2006]

Conforme se observa na Figura 19, diversos artefatos podem ser produzidos a fim de materializar a relação entre a disciplina *Análise e Design* e a fase de *Elaboração*. Todavia, será concentrada maior atenção no artefato "Classes de Análise", pois que estas "representam um primeiro modelo conceitual para elementos no sistema que possuam responsabilidades e comportamento" [Rational, 2006].

Diante de todo este contexto, então, pode-se concluir que os artefatos do IBM RUP a serem considerados para os fins desse trabalho serão aqueles que dizem respeito às disciplinas *Modelagem de Negócios*, *Requisitos e Análise e Design*. Por esta razão, os *Diagramas de Casos de Uso* (e seus respectivos *Diagramas de Atividade*), bem como o *Diagrama de Classes* serão tomados para fins da análise de similaridades conceituais entre *Gestão de Projetos* e *Análise Orientada a Objetos*, conforme detalhada no capítulo 4.

2.4 Conclusão

Conforme se observou pelo apresentado neste capítulo, a área de *Análise de Sistemas* evoluiu bastante suas abordagens, desde *Análise Estruturada* até a *Análise Orientada a Objetos*. É interessante notar que as abordagens iniciais em *Análise de Sistemas*, como a *Estruturada* e a *Essencial*, ressaltam uma visão mais ampla e integradora, as quais permitem que se desenvolva um sistema internamente coerente e coeso, seja pelo uso de *DFD Nível 0* (abordagem *top down*), seja pelo uso do *DFD Particionado por Eventos* (abordagem *bottom-up*).

Por outro lado, com a passagem para o contexto de *Análise Orientada a Objetos*, percebe-se que o número e a complexidade dos documentos formais entregues ao final de cada fase aumentaram significativamente durante este processo evolutivo. É indiscutível que a finalidade de tais documentos, agora chamados "artefatos", é prover um número suficiente e integrado de visões (percepções) detalhadas do sistema à equipe de desenvolvimento. Todavia, com as subseqüentes evoluções da UML,

bem como das metodologias que a utilizam, parece ter havido proliferação excessiva de artefatos, o que aponta na direção de uma provável perda da noção de que alguns deles deveriam servir como forma de levantamento de requisitos junto aos clientes. Afinal, vale lembrar que o modelo da realidade, delineado a partir daqueles requisitos, tem de ser compreendido por todas as partes envolvidas e não somente pela equipe de desenvolvedores. Como referiram Booch, Rumbaugh e Jacobson [2005, p.89], há que se ter em mente que o conceito de modelo diz respeito a “uma simplificação do mundo real, de forma que se possa entender melhor o sistema que está a ser desenvolvido”.

Um outro aspecto ainda decorrente do número crescente de artefatos a cada nova versão da UML e do RUP é que tal provocou uma fragmentação do modelo em inúmeros artefatos, o que pode dificultar a percepção do sistema como um todo, além, naturalmente, da comunicação entre desenvolvedores e clientes. A consequência natural e indesejada de todo este contexto, coerente com os dados estatísticos apresentados e discutidos nos capítulos iniciais deste trabalho, foi o aumento da possibilidade de elaboração e inserção de requisitos desnecessários ou defeituosos. Para além disso, a não percepção do todo potencia o negligenciamento de requisitos importantes que propiciariam a integração entre as diversas atividades do sistema.

Assim, diante de todo esse contexto, atualmente inerente à área de desenvolvimento de sistemas informatizados, há que se buscar novos métodos e técnicas que propiciem à análise orientada a objetos a recuperação dessas capacidades de facilidade de comunicação com o cliente quando do levantamento de requisitos, bem como de sua visão integradora entre as diversas partes do sistema, atualmente diluída em inúmeros artefatos. Desta forma, conforme já adiantado na abertura desse estudo, espera-se que alguns documentos típicos de definição de âmbito em gestão de projetos possam auxiliar nesse processo evolutivo.

3. Conceitos relevantes em Gestão de Projetos

3.1 Introdução

A fim de se alcançarem os objetivos deste estudo, faz-se necessário, naturalmente, um adequado entendimento sobre alguns dos principais conceitos em Gestão de Projetos, bem como das principais técnicas e documentos relacionados ao delineamento do âmbito de um projeto.

De fato, existem, no mercado, diversas metodologias referentes a esse tema, cada uma vinculada a uma organização internacional específica. Neste sentido, podem ser citadas, a título de exemplo, importantes iniciativas, como a *IPMA Competence Baseline* [IPMA, 2012], preconizada pela *International Project Management Association (IPMA)*; o *Ziel Orientierte Projekt Planung – ZOPP* (Planejamento de Projetos Orientado a Objetivos) [ZOPP, 2012], incentivado pela Agência Alemã de Cooperação Técnica (GTZ); o *Results Based Management -RBM* [CIDA, 2012], em uso pela Agência Canadense de Desenvolvimento Internacional; ou, ainda, o inglês *Projects IN Controlled Environments - PRINCE2* [APM Group, 2012], entre outras.

No entanto, uma das mais utilizadas internacionalmente é aquela elaborada pelo *Project Management Institute – PMI*, criado em 1969 nos EUA, e que conta, atualmente, com mais de 600.000 profissionais associados, em mais de 180 países [PMI, 2012]. Sua abordagem é veiculada a partir de seu conhecido guia de boas práticas, o *Project Management Body of Knowledge – PMBoK*, que, conforme já citado no capítulo 1, encontra-se na sua 4ª edição. Tais características de tempo e volume de uso (maturidade), associadas a sua notável amplitude internacional (neutralidade cultural), fazem com que esta metodologia seja a mais indicada para a revisão dos conceitos que ora se analisam.

Deste modo, de acordo com Miguel [2009, p.71], e em consonância com o guia PMBoK [2008, p.19], os processos de gerenciamento de projetos são classificados em cinco fases, conforme Figura 20 e descritos a seguir:

- **Fase de Iniciação**, contem os processos destinados a definir e autorizar um novo projeto ou uma fase de um projeto existente;
- **Fase de Planejamento**, contem os processos destinados a definir o âmbito do projeto, refinar objetivos e definir o curso de ação necessário para alcançar os objetivos e o âmbito para o qual o projeto foi iniciado;

- **Fase de Execução**, contem os processos destinados a integrar pessoas e outros recursos para executar o trabalho definido no Plano de Projeto;
- **Fase de Monitoramento/Controle**, contem os processos destinados a monitorar, rever e regular o progresso e o desempenho do projeto, identificar áreas em que seja necessário efetuar alterações ao Plano de Projeto e executar estas alterações; e
- **Fase de Encerramento**, contem os processos destinados a concluir todas as atividades ao longo de todas as fases para encerrar formalmente o projeto ou uma fase do projeto.

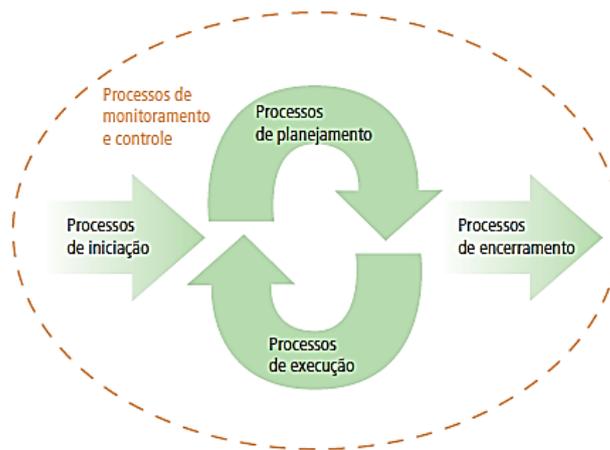


Figura 20 - Relacionamento entre as fases em gestão de projetos, segundo o PMBoK.
Arantes, Anselmo, Senise e Sabinelli [2008, p. 11]

Durante todo o ciclo de vida do projeto, esses processos interagem entre si trocando informações, de forma que um processo produz entradas para a execução de outro(s) processo(s) ou, ainda, entregas (*deliverables*). Cada processo ocorre no mínimo uma vez e, dentro destes, são produzidos diversos documentos de forma a auxiliar no gerenciamento do projeto [Miguel 2009, p.62], conforme ilustrado na Figura 21:



Figura 21 - Processos sobrepostos de gestão de projetos
Arantes, Anselmo, Senise e Sabinelli [2008, p. 11]

Neste momento, tendo em vista os objetivos deste estudo, cumpre destacar a elevada relevância das fases de *Iniciação* e de *Planejamento*, as quais, apesar de serem relativamente curtas, têm enorme impacto no sucesso (ou insucesso) final de um sistema desenvolvido, conforme já evidenciado pelas estatísticas do *Standish Group* em Eveleens e Verhoef [2010, p.31]. Desta forma, documentos produzidos nestas fases, como Termo de Abertura de Projetos, Declaração de Âmbito, *Work Breakdown Structure* (WBS), Matriz de Responsabilidades e Diagrama de Precedência [PMBok 2008, p.43], irão tratar de aspectos de interesse deste estudo. Conforme se observará no capítulo 4, tais documentos, analisados em detalhes a seguir, serão importantes para o estabelecimento da conexão desejada.

3.2 Termo de Abertura do Projeto

A fase de iniciação do projeto é composta por dois grandes processos: “Desenvolvimento do Termo de Abertura (área de conhecimento da integração) e Identificação dos *Stakeholders* (área de conhecimento da comunicação)” [Miguel 2009, p. 96]. De acordo com este mesmo autor, a elaboração do termo de abertura tem a finalidade principal de documentar os requisitos do negócio, justificar o projeto, compreender os requisitos reais do cliente e do novo produto, serviço ou resultado que se pretende desenvolver para atender a estas necessidades.

Este documento reconhece e comunica o âmbito aprovado do projeto aos gerentes funcionais e sua equipe, estabelecendo as responsabilidades e, ainda, fornece ao gerente de projetos a autoridade que necessita para que possa empregar recursos da organização para andamento das atividades. De acordo com Dinsmore e Cavaliere [2008, p.20], um termo de abertura deve incluir, entre outros, direta ou indiretamente: a) o objetivo e justificativa do projeto; b) a necessidade de negócio, descrição de alto nível do projeto ou requisitos do produto que o projeto deverá contemplar; c) os requisitos que satisfazem as necessidades e expectativas dos clientes, *sponsor* e outros interessados pelo projeto; d) o gestor do projeto designado e respectivo nível de autoridade; e) a designação do gerente do projeto; f) o cronograma sumarizado por meio de marcos; g) as influências dos *Stakeholders*; as organizações funcionais e respectivas participações; h) as premissas e restrições organizacionais, ambientais e externas e, i) orçamento sumarizado.

Ainda para os autores, algumas organizações têm desenvolvido o Termo de Abertura do Projeto de maneira mais detalhada, acrescentando o âmbito e objetivos do projeto, especificações, EAP (Estrutura Analítica do Projeto); *Timing* e plano de gastos e o plano de gerenciamento com os requisitos de recursos e carga horária; currículos dos colaboradores – chave, relações organizacionais e de estrutura, matriz de

responsabilidades, apoio necessário de outras organizações, políticas e procedimentos do projeto, plano de gerenciamento de mudança e aprovação da gerência dos demais documentos.

Depois de realizada a aceitação do Termo de Abertura, este torna-se o primeiro documento oficial do projeto. No entanto, é de grande importância ressaltar que, o termo de abertura deve ser elaborado de uma forma flexível, de maneira que seja permitido realizar mudanças que provavelmente acontecerão durante o ciclo de vida do projeto [Dinsmore e Cavellieri 2008, p.21].

3.3 Declaração de Âmbito

Elaborar a declaração de âmbito é um passo de suma importância, pois neste documento constará uma definição preliminar, em alto nível, do projeto. De modo geral, este documento deverá conter as características do produto ou serviço do projeto, os principais resultados e os seus objetivos. Esta declaração tem a finalidade de tratar e documentar os requisitos do projeto e da sua entrega, bem como os requisitos do produto, os limites do projeto, os métodos de aceitação e o controle de alto nível do âmbito [PMBok 2008, p.353].

Para se alcançar tal finalidade, faz-se necessário utilizarem-se *inputs* como o Termo de Abertura do Projeto, em conjunto com outros típicos da fase de iniciação. Conforme destacam Dinsmore e Cavellieri [2008, p. 22] “A confecção da declaração do escopo preliminar do projeto é baseada em técnicas, como análise do produto, análise de custo *versus* benefício, identificação de alternativas e avaliações especializadas, a partir de informações fornecidas pelo patrocinador (iniciador). O refinamento desse documento, pela equipe do projeto, origina a declaração do escopo”.

Segundo aqueles autores, tal declaração define o trabalho a ser realizado, funciona como auxílio à gerência na tomada de decisões e, ainda, auxilia os interessados no projeto na compreensão do âmbito do mesmo. Como qualquer outro processo, a declaração do âmbito deverá ser monitorada e atualizada de forma que sempre reflita às mudanças que vierem a ocorrer durante o ciclo de vida do projeto.

Miguel [2010, p.87], afirma que uma declaração de âmbito possui como principais objetivos: “(1) Estabelecer uma comunicação entre todos os interessados no projeto acerca do objetivo do projeto, o que o projeto deverá produzir e entregar, as pessoas envolvidas, os prazos, custos, qualidade e como o projeto será gerenciado; (2) Fornecer uma *baseline* para medir o progresso e para decidir se o projeto está concluído; e (3) Fornecer o ponto de referência para decidir se os pedidos de alteração ao projeto são válidos”.

Ainda de acordo com Miguel [2010, p.88], este documento inclui os itens a seguir, diretamente ou por meio de referência a outros documentos, a saber:

- *Objetivos do projeto:* incluem os respectivos critérios de sucesso mensuráveis. Os projetos podem possuir uma grande variedade de objetivos empresariais, técnicos, de qualidade, de custo e de prazo;
- *Âmbito do produto a executar:* descrição das características do produto, serviço ou resultado que o projeto visa construir. Essas características são geralmente menos detalhadas nas fases iniciais e progressivamente mais detalhadas nas fases mais avançadas. Embora a forma e a substância das características possam variar, a descrição do âmbito deve proporcionar sempre suficiente detalhe para suportar o posterior planejamento do âmbito do projeto;
- *Fronteiras do projeto:* identificação daquilo que está incluído no projeto. Se houver a possibilidade de um qualquer interveniente no projeto poder assumir que um determinado produto, serviço ou resultado poderá ser considerado uma componente de projeto, então este ponto deve indicar explicitamente aquilo que está excluído do projeto;
- *Deliverables do projeto:* incluem quer os resultados que compreendem o produto ou serviço do projeto quer resultados subsidiários, como os relatórios e a documentação de gestão do projeto. Dependendo da descrição do âmbito do projeto, as entregas poderão ser descritas de uma forma muito detalhada ou apenas de uma forma resumida;
- *Crítérios de aceitação dos deliverables:* definição dos processos e critérios a serem usados para aceitação dos produtos que são concluídos;
- *Restrições e pressupostos do projeto:* deve ser feita uma descrição das restrições específicas do projeto, associadas ao seu âmbito e que limitam as opções da equipe (por exemplo, um orçamento predefinido ou datas impostas). Quando um projeto é realizado sob contrato, as cláusulas contratuais constituem geralmente restrições. Pressupostos são fatores que, para efeitos de planejamento, são considerados verdadeiros ou reais pela equipe de projeto (por exemplo, assumir-se que um equipamento será entregue numa determinada data, para que se possam planear as ações seguintes). Os pressupostos devem ser listados e o impacto que o não cumprimento dos mesmos terá no projeto pode ser evidenciado;
- *Milestones do projeto:* identificação dos principais marcos do projeto. Estes marcos são, regra geral, datas impostas pelo cliente ou *sponsor* e são tratados como restrições ao cronograma do projeto.

Do que se observa, então, pelos itens apresentados por aquele autor, a Declaração de Âmbito contém uma série de aspectos delimitadores do projeto a ser desenvolvido, tanto de caráter do produto em si, quanto dos recursos necessários para levá-lo a termo e suas restrições.

3.4 Work Breakdown Structure – WBS

A *Work Breakdown Structure – WBS*, ou Estrutura Analítica do Projeto – EAP, que é a sua tradução usual para o português, “é um agrupamento orientado a produtos de elementos do projeto e define o escopo [âmbito] total do trabalho” [Dinsmore e Cavelieri 2008, p.46]. Desta forma, no que diz respeito a gerenciamento de projetos, o WBS deverá conter todos os seus requisitos, de tal forma que o que não estiver neste documento, não será atendido pelo projeto. O objetivo deste documento é assegurar que a equipe envolvida realizará somente o trabalho necessário para atingir o objetivo definido.

A WBS evolui com o projeto e possibilita a identificação e organização das unidades lógicas de trabalho a serem gerenciadas, denominadas pacotes de trabalho [Miguel 2009, p.124]. Estes estão organizados em uma estrutura hierárquica, contendo os diversos itens a serem cumpridos (ou *deliverables*) para se atingir os objetivos finais. Desta forma, é utilizado como base para planejar, organizar e controlar todo o trabalho a ser feito.

De acordo com o *Project Management Institute* [2006, p. 11], ao se concluir a elaboração de uma WBS de maneira correta, esta deverá ter alcançado os seguintes objetivos:

- Definir a hierarquia entre os produtos a serem entregues (*deliverables*);
- Refletir a definição de todo o trabalho necessário para se alcançar um objetivo final e/ou os produtos a serem entregues;
- Prover uma representação gráfica ou um delineamento textual do âmbito do projeto;
- Prover um *framework* para todos os produtos a serem entregues por todo o ciclo de vida do projeto;
- Prover uma forma de integrar e avaliar o cronograma e o desempenho dos custos;
- Facilitar a distribuição dos recursos;
- Facilitar a comunicação e análise dos dados de progresso e situação; e
- Prover um *framework* para especificar objetivos de desempenho.

Para alcançar esses objetivos, Dinsmore e Cavelieri [2008, p.46] apontam os principais passos para elaboração de uma WBS:

1. Colocar em primeiro nível (nível 0) da WBS o nome do projeto;
2. Colocar em segundo nível (nível 1) as fases que estabelecem o ciclo de vida do projeto. Este nível também é conhecido como “primeiro nível de decomposição”;
3. Acrescentar um elemento, no segundo nível, para conter os *deliverables* necessários ao gerenciamento do projeto;
4. Identificar os subprodutos necessários para que seja alcançado o sucesso do projeto em cada fase;
5. Verificar, para cada subproduto, se as estimativas de custo e tempo, assim como a identificação de riscos e a atribuição de responsabilidade para a execução do mesmo, podem ser desenvolvidas neste nível de detalhe.
6. Rever e refinar a WBS até que o planejamento do projeto possa ser completado.

De acordo com Miguel [2010, p. 89], vários processos podem ser utilizados para a criação da hierarquia da WBS, conforme exemplificado na Figura 22. Para este mesmo autor, a **Fase** “refere-se ao nível mais elevado de atividades, como análise, desenho, codificação, etc”. Ainda segundo o autor, “o trabalho de nível mais elevado é designado por actividade e estas são constituídas por tarefas”. A Figura 22 ilustra este entendimento.

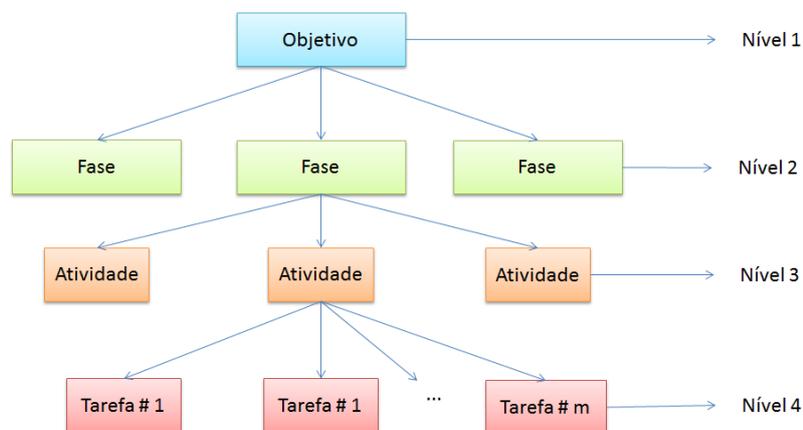


Figura 22 - Hierarquia de termos de uma WBS
Miguel [2010, p. 89]

De todo o modo, ao fundirem-se as percepções de Dinsmore e Cavelleri [2008] e Miguel [2010], é suposto que, ao término desse processo, o desenvolvedor obtenha um WBS semelhante ao ilustrado pela Figura 23 abaixo.

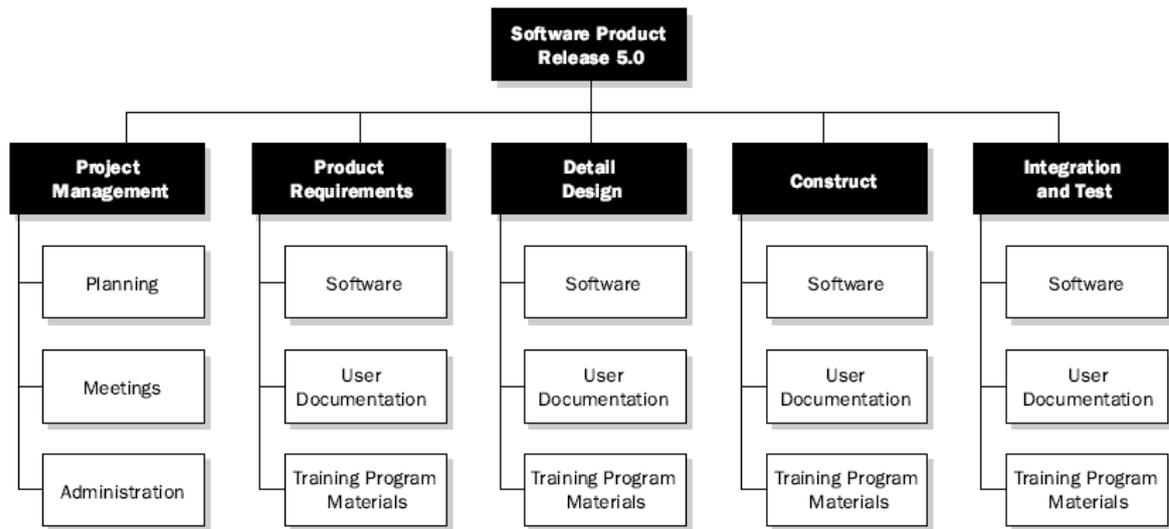


Figura 23 - Exemplo de decomposição com principais entregas
 PMBoK [2008, p. 119]

Do que se observa, então, acerca da WBS e suas características intrínsecas, tal técnica constitui-se numa ferramenta robusta e simples, a qual se preocupa única e exclusivamente com o levantamento e delimitação de âmbito de projetos. Deste modo, nota-se que a mesma não lida com outras questões relevantes no desenvolvimento de um projeto, como, por exemplo, a distribuição das atividades entre os diversos membros da equipe. Para tal fim, outras técnicas são utilizadas.

3.5 Matriz de Responsabilidades

Durante o ciclo de vida de um projeto, todos os envolvidos possuem funções as quais devem ser desempenhadas. A Matriz de Responsabilidades trata-se de uma ferramenta a qual é desenvolvida durante a fase de Planejamento. Seu objetivo fundamental é correlacionar, para cada atividade ou tarefa, o responsável pela sua execução [Dinsmore e Cavelleri, 2008]. Para os mesmos autores, embora a responsabilidade do projeto como um todo seja do gerente, no que diz respeito a qualquer evento que venha a acontecer durante o seu desenvolvimento, faz-se importante salientar que, sob a ótica de cada atividade individual, “a responsabilidade sobre esta é da pessoa diretamente envolvida na mesma” [idem, p.145]. Tal entendimento nos remete à importância de uma definição concisa dos papéis e das responsabilidades quando da atribuição dos mesmos.

De acordo com Taylor [2008, p.130], este documento, ao fazer um mapeamento entre as atividades a serem executadas e os membros (ou grupos) da organização, “provê uma visão rápida e fácil de quem é primariamente responsável por cada componente do projeto e quem deve dar suporte”.

Dinsmore e Cavellieri [2008, p.145] consideram a Matriz de Responsabilidades uma das principais técnicas utilizadas pelos gerentes de projeto para garantir uma boa comunicação entre todos os envolvidos no projeto. A Tabela 2 ilustra um exemplo desta ferramenta, onde as linhas representam as principais responsabilidades e as colunas representam os responsáveis pelas mesmas. Apresentam-se ainda, neste exemplo, o nível de responsabilidade de cada integrante da equipe com relação a cada atividade.

H – Homologa V – Valida P – Participa R – Responsável

| Fase/Pessoa | José | Carlos | André | Ana | Paulo |
|---------------------------|------|--------|-------|-----|-------|
| Definição de necessidades | H | V | P | R | |
| Design de solução | H | | R | P | P |
| Análise detalhada | H | | P | | R |
| Desenvolvimento | H | V | | P | R |
| Testes | H | | | R | P |

Tabela 2 - Matriz de responsabilidades
Dinsmore e Cavellieri [2008, p. 145]

O Guia PMBoK [2008, p.438], por sua vez, estabelece que a Matriz de Responsabilidades (MR) é uma “estrutura que relaciona o *organograma* do projeto com a *estrutura analítica do projeto* [WBS] para ajudar a garantir que cada componente do escopo de trabalho do projeto seja atribuído a uma pessoa responsável”. Nesse sentido, tal definição está alinhada àquela apresentada por Dinsmore e Cavellieri [2008]. Todavia, o referido Guia é mais flexível do que aqueles autores, pois ressalta que “em projetos maiores, é possível desenvolver as MRs em vários níveis. Por exemplo, uma MR de alto nível pode definir os grupos ou unidades da equipe do projeto responsáveis pelos componentes da EAP [WBS], enquanto MRs de nível mais baixo são usadas dentro do grupo para atribuir funções, responsabilidades e níveis de autoridade para atividades específicas” [PMBoK 2008, p.221].

A partir deste último entendimento, a Matriz de Responsabilidades torna-se uma técnica apropriada tanto para a alta gerência, como para os gerentes intermediários. Para além disso, ao permitir a troca do nome das pessoas pelo nome das unidades de equipe do projeto, torna a matriz mais impessoal e, conseqüentemente, mais duradoura ao longo do tempo, pois os membros de uma equipe podem ser alterados a qualquer momento e por diversas razões.

3.6 Diagrama de Precedência

Outra questão relevante no universo de Gestão de Projetos diz respeito ao sequenciamento das atividades a serem executadas, de forma que as mesmas ocorram não apenas na ordem certa, mas, ainda, com o melhor aproveitamento possível do tempo. Por outras palavras, não se trata apenas de se colocar em fila as tarefas levantadas, mas, também, de se determinar quais destas tarefas podem ser executadas em simultâneo. Nesse sentido, a questão é se descobrir as atividades que devem ser executadas primeiramente para que outras, suas dependentes, possam ser executadas subsequentemente.

Segundo o Guia PMBoK [2008, p. 221], a técnica mais indicada, para estes casos, é o método do Diagrama de Precedência, no qual “a unidade básica de análise é a atividade” [Miguel 2009, p. 134]. Neste tipo de diagramação, cada atividade é representada por um retângulo, chamado de “nó”, os quais são conectados por setas que representam a dependência entre eles, denotando, também, a relação antecessor/sucessor entre os mesmos [idem]. Note-se, ainda, que cada atividade deverá estar contida em um único nó e que sua realização viabilizará a execução dos nós seguintes. A Figura 24 ilustra um exemplo de tal diagrama.

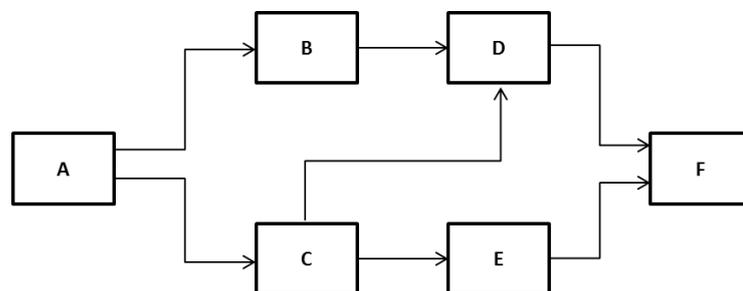


Figura 24 - Método do Diagrama de Precedência
Miguel [2010, p.134]

O Diagrama de Precedência, como conceito, apresenta relevância para os objetivos deste estudo, tendo em vista que ele sequencia as várias atividades (ou responsabilidades) a serem executadas pelos diversos envolvidos no uso de um sistema a ser desenvolvido. Neste sentido, esta ferramenta mostra potencial como um auxílio a uma melhor compreensão e organização da WBS, bem como na distribuição das responsabilidades.

3.7 Conclusão

O presente capítulo apresentou a maturidade da área de gerenciamento de projetos, especificamente no que diz respeito à questão da definição de âmbito. De fato, tal está muito relacionada à aplicação extensiva do Guia PMBoK, o qual sugere a utilização sistemática de várias técnicas que facilitam a obtenção e gestão das atividades necessárias à execução de um projeto.

Deste modo, o capítulo apresentou os principais instrumentos utilizados em Gestão de Projetos para levantamento e delineamento de atividades inerentes ao âmbito de um projeto. Dentre estes, foram destacados alguns produzidos nas fases de Iniciação e Planejamento, os quais têm o objetivo de estabelecer os contornos do que deve ser feito para assegurar que o cliente receberá um produto com todas as funcionalidades solicitadas e, conseqüentemente, com um nível maior de qualidade.

Neste sentido, pode-se observar que determinadas abordagens aqui apresentadas, nomeadamente, a Declaração de Âmbito, o Termo de Abertura do Projeto, a *Work Breakdown Structure* (WBS), a Matriz de Responsabilidades e o Diagrama de Precedência, cumprem com objetividade a tarefa de tornar claras as diversas atividades a serem executadas, ao facilitar a comunicação com o cliente, bem como ao garantir que todos os envolvidos (clientes e desenvolvedores) tenham uma visão global dos requisitos do sistema a ser desenvolvido.

Tais documentos podem, presumivelmente, por conta de tais características, cobrir as falhas de integração e visão global percebidas na Análise Orientada a Objetos, conforme explicadas na conclusão do capítulo anterior. Todavia, a simples transposição dos documentos da área de Gestão de Projetos, acima citados, para a área de Análise Orientada a Objetos irá requerer adequações aos mesmos, afinal trata-se de áreas de conhecimento distintas. Para além disso, ainda ao se tentar fazer tal transposição, haverá também a questão do ponto de vista (enfoque) do profissional que elabora tais documentos, pois, no primeiro caso, ele tenderá a tratar tais documentos como ferramentas de apoio à administração do desenvolvimento, ao passo que, no segundo caso, ele deverá elaborar os mesmos única e exclusivamente pelo enfoque dos requisitos do sistema.

De todo o modo, pelo que foi apresentado neste capítulo, percebem-se algumas semelhanças entre documentos de definição de âmbito em Gestão de Projetos com artefatos equivalentes na área de Análise Orientada a Objetos, fato este que será explorado em maiores detalhes no capítulo 4 deste estudo.

4. Análise de similaridades conceituais e proposta de processo para uso de documentos de Gestão de Projetos no delineamento de requisitos em AOO

4.1 Introdução

O embasamento teórico relevante, conforme apresentado nos capítulos 2 e 3, procurou fornecer uma compreensão seletiva das técnicas, tanto de Análise Orientada a Objetos, quanto de Gestão de Projetos, que lidam com o levantamento de requisitos, de forma a permitir a análise de possíveis similaridades conceituais entre ambas.

Todavia, antes de se iniciar tal análise, há um aspecto inerente a ambas que pode causar alguma confusão e este refere-se ao enfoque subjacente de cada uma delas. De fato, a Análise Orientada a Objetos (AOO), enquanto análise de sistemas, trata estes por meio de suas funcionalidades a implementar (requisitos), ao passo que a Gestão de Projetos, conforme o nome indica, procura olhar o mesmo não somente em termos de âmbito (requisitos), mas, principalmente, em termos de gestão das atividades necessárias ao seu desenvolvimento. Assim, por exemplo, os analistas de sistemas não se preocupam, por definição [Rational, 2006], com a contratação dos programadores necessários ou, ainda, se os salários destes estão sendo pagos em dia ou não. Inversamente, um gerente de projetos, também por definição [PMBok, 2008], tem a obrigação de se preocupar com aspectos desta natureza. Não por acaso, a instituição *Project Management for Development Organizations* alerta que, para além da necessidade de os gerentes de projetos possuírem qualificações técnicas, também se faz necessário que estes tenham “boas habilidades de gestão, tais como: comunicação, planejamento, negociação, *coaching*, tomada de decisão e liderança” [PM4DEV 2007, p.75].

Esta sutil, porém importante, diferença tem que ser levada em conta ao se tentar utilizar os documentos de definição de âmbito em Gestão de Projetos no universo de desenvolvimento de sistemas orientados a objetos, pois que, de outro modo, acabaríamos por inserir, no sistema do cliente, uma série de requisitos de gestão do desenvolvimento do projeto.

4.2 Análise de similaridades conceituais

Ao se iniciar a análise das similaridades pretendidas entre as dimensões Gestão de Projetos e AOO, há que se ter em mente que se está a focalizar as fases iniciais de ambas. Neste sentido, ao tomarmos as

metodologias consideradas em cada uma delas (PMBok e RUP, respectivamente), tal equivale a dizer que, na dimensão Gestão de Projetos, há que se buscar conexões conceituais entre as suas fases de *Iniciação* e de *Planejamento*, e as fases de *Iniciação* e *Elaboração* da dimensão Análise Orientada a Objetos, uma vez que tais tratam das atividades relativas à elucidação e delineamento dos requisitos/âmbito a serem atendidos, conforme percebidos e acordados com o cliente.

Deste modo, é razoável afirmar que a estratégia de análise de similaridade conceitual entre as duas dimensões deverá ser procedida a partir dos documentos/artefatos que são produzidos nas referidas fases. Naturalmente, ao se tentar transpor documentos de gestão de projetos para a área de desenvolvimento de sistemas orientados a objetos, estes deverão sofrer uma adequação de sua forma e/ou conteúdo, tendo em vista seus diferentes enfoques, conforme anteriormente discutido.

Neste sentido, na dimensão *Análise Orientada a Objetos*, apesar de terem sido apresentados diversos artefatos capítulo 2, constatou-se que os de efetivo interesse para os objetivos deste estudo são os Diagramas de Casos de Uso, Diagramas de Atividades e o Diagrama de Classes. Por sua vez, na dimensão *Gestão de Projetos*, conforme apresentado no capítulo 3, são de interesse os seguintes documentos: Declaração de Âmbito, Termo de Abertura de Projeto, *Work Breakdown Structure* (WBS), Matriz de Responsabilidades e Diagrama de Precedência. A Figura 25, a seguir, apresenta, de forma esquematizada, a estratégia geral de análise, até este ponto:

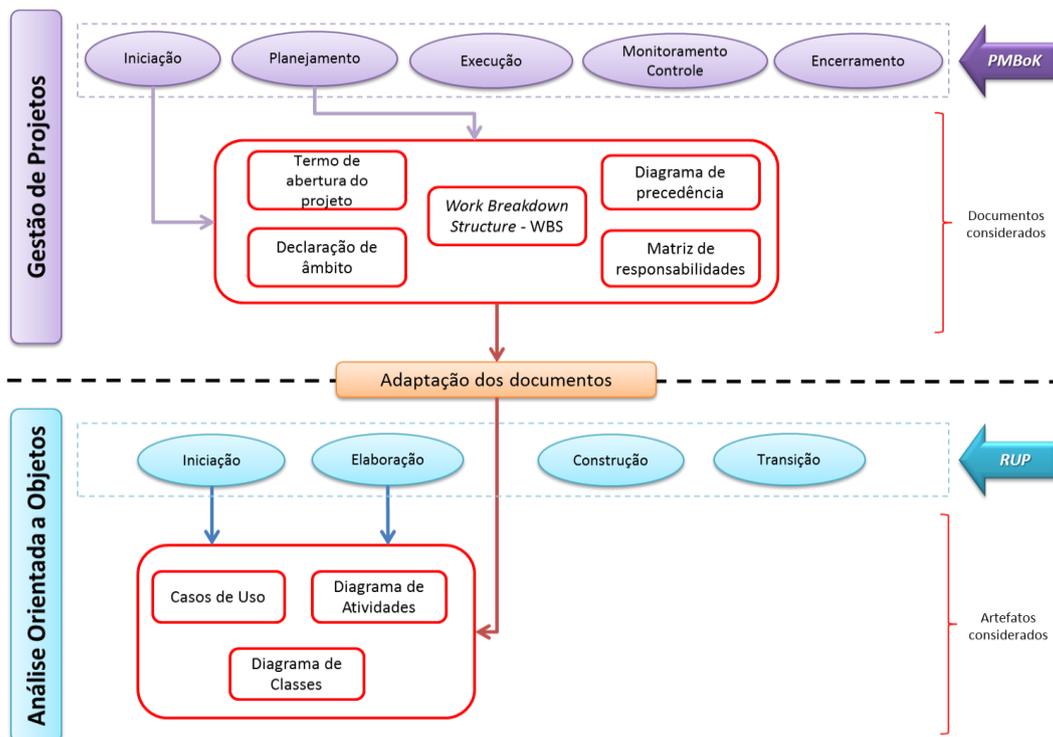


Figura 25 - Estratégia geral de análise de similaridades, vista a partir das fases de cada dimensão

Ocorre que, segundo Boehm [1996], qualquer projeto de sistema de informação deverá responder sete questões cruciais, no que ficou conhecido como W5H2 [Silva e Videira 2001, p.30], a saber:

1. Por que (*why*): por que o sistema vai ser desenvolvido?
2. O que (*what*): o que vai ou deve ser feito?
3. Quando (*when*): quando é que vai ser feito?
4. Quem (*who*): quem é o responsável por uma função?
5. Onde (*where*): onde se encontram localizadas na organização (as responsabilidades)?
6. Como (*how*): como é que vai ser feito, tecnicamente e gerencialmente?
7. Quanto (*how much*): quanto de cada recurso será necessário?

Desta forma, ao se observar atentamente os documentos e artefatos considerados em cada fase relevante e em cada dimensão, conforme mostrados na Figura 25, percebe-se que tais concentram-se principalmente sobre três daquelas questões fundamentais:

- **O quê** - este referindo-se aos requisitos a serem cumpridos no desenvolvimento do sistema;
- **Quem** - este referindo-se às entidades externas ao sistema que têm (ou terão) demandas informacionais sobre o mesmo;
- **Como** - este, por sua vez, diz respeito ao encadeamento lógico entre as atividades que o sistema deve realizar a fim de atender a uma demanda informacional.

Nesse sentido, a Figura 26, abaixo, ilustra a possibilidade de similaridades conceituais entre documentos e artefatos, a partir das questões acima citadas.



Figura 26 - Análise de similaridades conceituais entre Gestão de Projetos e AOO

Do que se percebe, então, a partir da referida figura, existem documentos (Gestão de Projetos) e artefatos (Análise Orientada a Objetos), que modelam aspectos similares de um problema. Tais

similaridades de objetivos/finalidades podem ser detalhadas conforme apresentadas na Tabela 3, a seguir:

| <i>DIMENSÃO</i> | <i>DOCUMENTO/ARTEFATO</i> | <i>O QUÊ?</i> | <i>QUEM?</i> | <i>COMO?</i> |
|--------------------------------|---------------------------------|---------------|--------------|--------------|
| Gestão de Projetos | Termo de Abertura de Projeto | X | | |
| | Declaração de Âmbito | X | | |
| | <i>Work Breakdown Structure</i> | X | | |
| | Matriz de Responsabilidades | X | X | |
| | Diagrama de Precedência | X | | X |
| Análise Orientada a Objetos | Diagrama de Casos de Uso | X | X | |
| | Diagrama de Classes | X | X | X |
| | Diagrama de Atividades | X | X | X |

Tabela 3 - Relação entre documentos e artefatos, a partir das questões centrais

Conforme se observa na tabela anterior, há uma possibilidade de estabelecimento de pontos de conexão conceituais e finalísticos entre as áreas de Gestão de Projetos e de Análise Orientada a Objetos, a partir dos documentos e artefatos considerados, no que diz respeito à definição de âmbito/requisitos. Em outras palavras, ao considerar-se cada dimensão a partir dos documentos/artefatos analisados, observam-se as similaridades conceituais buscadas.

Assim, ao analisar-se a Tabela 3, percebe-se que, a partir das similaridades conceituais observadas, os documentos de gestão de projetos considerados podem dar contribuições para a elaboração de artefatos em análise orientada a objetos. Assim, partindo-se das conexões conceituais conforme ilustradas na Figura 27, observa-se que o conceito “Quem”, alimentado pela Matriz de Responsabilidades, pode ajudar a compor os Diagramas de Casos de Uso, de Classes e Atividades. Por sua vez, o conceito “O que”, alimentado pelos documentos Declaração de Âmbito, Termo de Abertura de Projeto, WBS, Matriz de Responsabilidades e Diagrama de Precedência, também podem ajudar a compor os Diagramas de Casos de Uso, de Classes e de Atividades. Por fim, o conceito “Como”, a partir do Diagrama de Precedência, tem potencial para auxiliar na elaboração dos Diagramas de Classes e de Atividades.

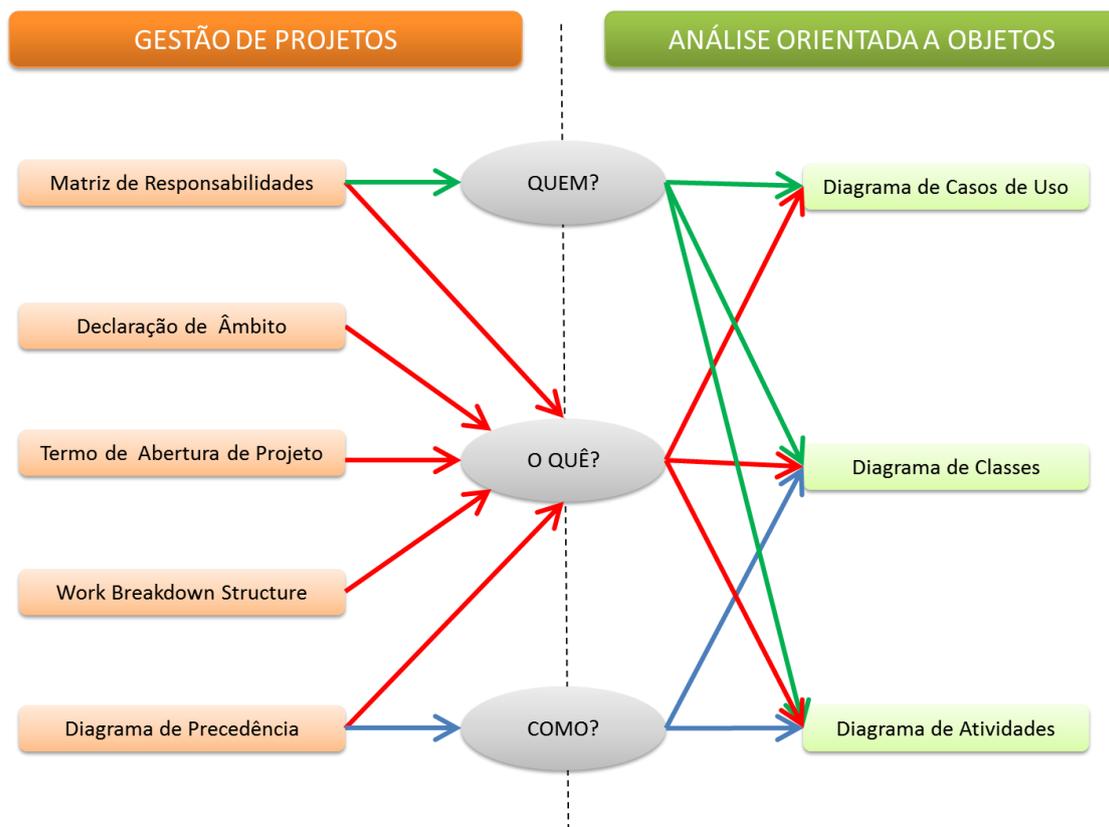


Figura 27 - Estratégia inicial de conexões conceituais entre documentos e artefatos

No entanto, toda esta estratégia inicial necessitou de ajustes, por conta dos argumentos apresentados na introdução deste capítulo. Assim, uma das consequências foi a necessidade de se retirar o Termo de Abertura de Projeto (TAP) do rol de documentos de definição de âmbito, em Gestão de Projetos, passíveis de uso para os fins deste estudo. Tal deveu-se ao fato de que, conforme apresentado no item 3.2, aquele é um documento rico em informações de caráter administrativo, porém deficiente sob a ótica de âmbito ou de objetivos do sistema, o que o torna pouco útil para o levantamento dos requisitos de um sistema a ser desenvolvido. A única exceção a esta última afirmação ficaria por conta dos requisitos ali declarados. No entanto, os mesmos podem perfeitamente ser inseridos no contexto do item “Âmbito do produto a executar”, o qual já faz parte de uma Declaração de Âmbito, conforme vista no tópico 3.3, e que é mais abrangente.

Da mesma forma, o Diagrama de Precedência, conforme apresentado no item 3.6 deste estudo, ao ordenar as atividades a serem executadas, mostrava potencial de uso para o delineamento de requisitos no universo de desenvolvimento de sistemas, porém tal não se confirma. Isto decorre do fato de serem diferenciadas as finalidades do uso da precedência em cada uma das dimensões. Este diagrama, no

contexto de Gestão de Projetos, trata de *ordenar as atividades que viabilizam o alcance de seu âmbito*. Por outro lado, num contexto de delineamento de requisitos de um sistema, este tentaria *ordenar os requisitos em si*, o que nem sempre se verifica. Em realidade, tal precedência, no universo de desenvolvimento de sistemas, irá ocorrer principalmente ao nível dos algoritmos, os quais já são tratados por diagramas específicos como de Atividades e de Sequência. O resultado de todo este contexto foi a necessidade de se eliminar o Diagrama de Precedência como potencial ferramenta de uso na delimitação de requisitos em sistemas orientados a objetos.

Resumindo, então, o Termo de Abertura de Projetos e o Diagrama de Precedência tiveram de ser abstraídos da estratégia inicial, tendo em vista sua pouca utilidade para os objetivos deste trabalho, conforme discutido acima. Com isso, a estratégia de conexão final ficou simplificada, conforme se pode observar pela Figura 28, a seguir:

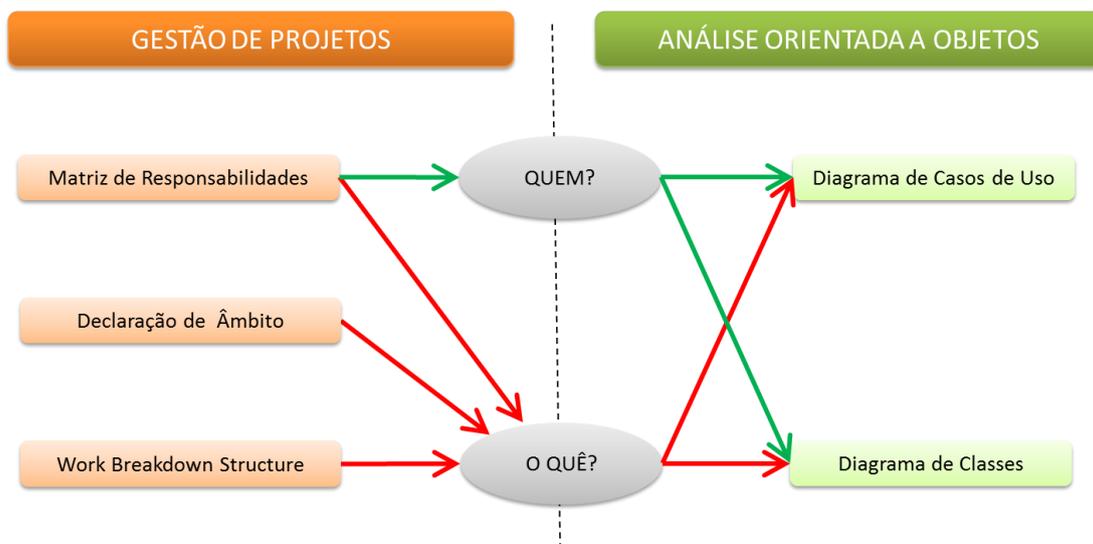


Figura 28 - Estratégia final de conexões conceituais entre documentos e artefatos

Agora, ao tomarmos em conta o que está representado na Figura 28, observa-se que os documentos de gestão de projetos considerados na estratégia final podem fornecer informações úteis para a elaboração de artefatos em análise orientada a objetos. Assim sendo, agora nota-se que o conceito “Quem”, alimentado pela Matriz de Responsabilidades, ajudará a compor os Diagramas de Casos de Uso e de Classes. Por sua vez, o conceito “O que”, alimentado pelos documentos Declaração de Âmbito, WBS e Matriz de Responsabilidades, ajudarão a compor os Diagramas de Casos de Uso e de Classes. Por fim, observe-se que, em decorrência da eliminação do Diagrama de Precedência, também foi retirada da estratégia inicial o suporte deste à elaboração do Diagrama de Atividades.

Assim, uma vez estabelecida essa estratégia final de conexão, foi possível elaborar uma proposta consistente para se executar a transposição de documentos entre as duas dimensões, em conformidade com os objetivos deste estudo.

4.3 Processo proposto

A presente proposta de processo de uso de documentos de Gestão de Projetos em Análise Orientada a Objetos toma por base a própria definição do que vem ser um *processo*. Nesse sentido, segundo o Dicionário Priberam da Língua Portuguesa [Priberam, 2012], tal termo pode ser entendido tanto como “método, sistema ou modo de fazer uma coisa”, quanto um “conjunto de manipulações para obter um resultado”. Em consonância com esse entendimento, o Dicionário Fundamental da Língua Portuguesa [Dicionário Universal 2009, p.552], afirma que um processo pode ser entendido como um “método”, sendo este, por sua vez, segundo a mesma fonte, um “processo racional que se segue para chegar a um fim” ou, ainda, um “conjunto de procedimentos técnicos e científicos”[p.478].

Assim, tendo por base tais entendimentos, as etapas do processo presentemente proposto foram elaboradas a partir das similaridades conceituais entre documentos e artefatos, conforme observados na estratégia final estabelecida no tópico anterior. Deste modo, então, ao se tomar em conta os documentos listados na Figura 28, observa-se que as seguintes etapas devem ser cumpridas a fim de se viabilizar a pretendida conexão entre as duas dimensões:

- Elaboração da Declaração do Âmbito
- Elaboração do WBS;
- Elaboração da Matriz de Responsabilidades;
- Elaboração dos Diagramas Casos de Uso iniciais; e
- Elaboração do Diagrama de Classes inicial.

O cumprimento de tais etapas viabilizará o aproveitamento dos referidos documentos, típicos de gestão de projetos, na definição de requisitos em análise de sistemas orientados a objetos. A execução dessas etapas, no entanto, irão requerer alguns cuidados, conforme apresentados a seguir.

4.3.1 *Elaboração da Declaração de Âmbito*

Conforme visto no tópico 3.3, há uma série de informações que são usualmente incluídas neste documento, quando utilizado no universo de gestão de projetos. No entanto, para o contexto de análise de sistemas, sugere-se a manutenção apenas dos seguintes aspectos, devidamente adaptados e elaborados a partir do ponto de vista do cliente:

- *Objetivos do sistema:* Descrição sucinta sobre o que trata o sistema.
- *Âmbito do sistema:* Descrição dos principais assuntos a serem cobertos pelo sistema. Não se recomenda entrar em detalhes específicos, mas, antes, que se informe os principais módulos que irão compor o sistema a ser entregue.
- *Fronteiras ambientais do sistema:* Esclarecimentos sobre em quais ambientes o sistema deverá operar. Neste sentido, há que se informar se o sistema será executado apenas na empresa, remotamente nas filiais, acesso móvel, acesso via *internet*, etc.
- *Restrições e pressupostos do sistema:* Apresentação de constrangimentos tecnológicos do sistema a ser desenvolvido (sistemas operacionais, base de dados, etc.), bem como outros intervenientes, como por exemplo, legislações a serem obedecidas, normas de qualidade, questões de gestão da segurança da informação, auditoria de sistemas, etc.

Para além destes aspectos, deve-se, ainda, inserir um item, imediatamente após "*Fronteiras ambientais do sistema*", que informe os tipos de utilizadores (atores) do sistema. A importância desta informação traduzir-se-á na elaboração da matriz de responsabilidades, em uma etapa posterior.

4.3.2 *Elaboração da WBS*

Para a elaboração da WBS, naturalmente, deverá ser mantida a sua forma original em termos de estruturação. Todavia, o enfoque de levantamento de suas atividades deverá ser ajustado de modo que as mesmas digam respeito somente ao uso do sistema. Quaisquer atividades típicas de administração do ciclo de vida do desenvolvimento deverão ser desconsideradas, conforme as razões apresentadas e discutidas no tópico 4.1. Deste modo, as atividades que irão compor a WBS serão os próprios requisitos funcionais do sistema, os quais podem ser obtidos de diversas formas, algumas das quais apresentadas no capítulo 2. Para além destas, sugere-se também o uso da Declaração de Âmbito definida na etapa anterior. Em termos práticos, recomenda-se:

- a) Colocar o nome do sistema no nível 0 da WBS;
- b) Colocar, no nível 1, os principais temas (assuntos), sob a ótica do cliente, que o sistema deverá gerir. Inclusivamente, sugere-se que os nomes destas atividades iniciem-se com o verbo “Gerir”;
- c) Para o nível 2, deve-se inserir os requisitos que, ainda sob a ótica do cliente, permitirão a execução das atividades inerentes à sua atividade-pai, que estará no nível 1.
- d) Caso haja necessidade de níveis inferiores, estes devem ser elaborados sempre sob a ótica do cliente. No entanto, recomenda-se não ir além do nível 3, pois corre-se o risco de se inserir requisitos não-funcionais no sistema a ser desenvolvido.
- e) Rever e refinar a WBS, junto ao cliente, até que esta esteja de pleno acordo com a sua estrutura.

Ao término desse processo, obter-se-á uma WBS que conterà todos os requisitos funcionais do sistema, pela ótica do cliente. A Figura 29, a seguir, exemplifica trecho de uma possível WBS para um Sistema de Gestão de Biblioteca hipotético.

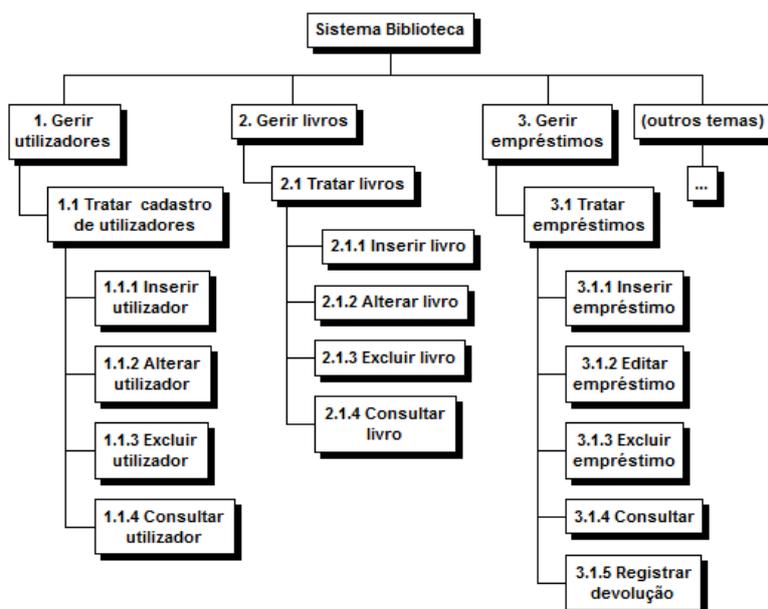


Figura 29 - Exemplo de trecho de WBS para um Sistema de Gestão de Biblioteca

4.3.3 *Elaboração da Matriz de Responsabilidades*

Uma vez elaborada a WBS, esta deverá conter, por princípio, todos os requisitos/atividades que serão passíveis de serem executados a partir do sistema. No entanto, conforme explicado no item 3.4 deste trabalho, a WBS não se destina a informar quem são os responsáveis por executar as atividades nela constantes. Posto de outra forma, aquele documento não contém nenhuma informação sobre os

perfis de acesso ao sistema, ou seja, as responsabilidades cabíveis a cada tipo de utilizador do sistema. Por outro lado, conforme apresentado e discutido no tópico 3.5, uma Matriz de Responsabilidades cumpre bem essa tarefa, pelo que é aplicável a este contexto.

Todavia, a fim de se elaborar a referida matriz sob o novo enfoque e objetivando cumprir a presente etapa, recomenda-se:

- Transpor toda a estrutura da WBS elaborada na etapa anterior, com todos os seus níveis e subníveis, para a matriz de responsabilidades. O nível 1 deverá ser incluído numa coluna chamada “Tema/Assunto”; o nível 2, numa chamada “Atividades”; por fim, o nível 3, numa coluna chamada “Responsabilidades”;
- Definir os atores, obtidos na Declaração de Âmbito, como “Papéis”, em colunas próprias; e
- Para cada responsabilidade definida na WBS, verificar, junto ao cliente, sua interação com cada papel e, caso aquele papel tenha responsabilidade sobre aquela atividade, tal fato deverá ser marcado na matriz; caso contrário, deixa-se o espaço em branco.

De modo esquemático, a Figura 30, abaixo, apresenta o procedimento para esta etapa.

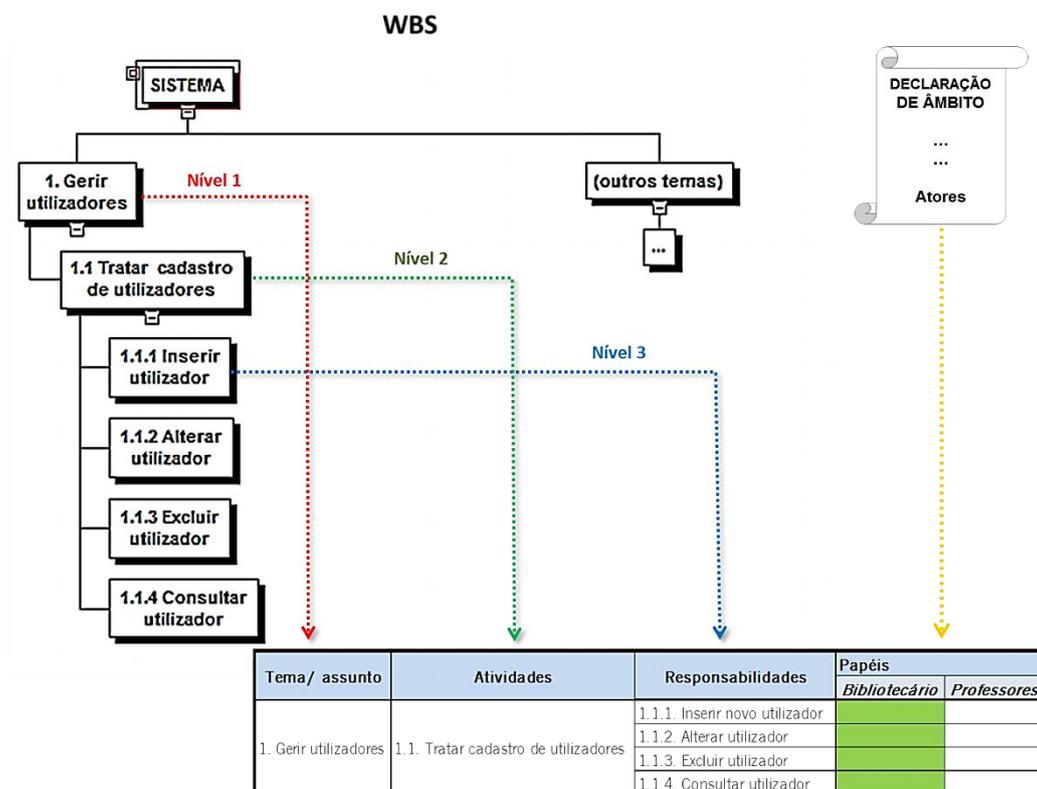


Figura 30 - Esquema de elaboração da Matriz de Responsabilidades, a partir da WBS e da Declaração de Âmbito

Uma vez cumprida esta etapa, torna-se viável utilizar esta matriz como auxílio à elaboração dos casos de uso correspondentes. A Figura 31, a seguir, exemplifica uma matriz com esta finalidade, um pouco mais completa, elaborada para os temas 1 e 2, a partir da WBS apresentada na Figura 29.

| <i>Sistema de Gestão de Biblioteca</i> | | | | | |
|--|--------------------------------------|--------------------------------|----------------------|--------------------|---------------|
| Tema/ assunto | Atividades | Responsabilidades | Papéis | | |
| | | | <i>Bibliotecário</i> | <i>Professores</i> | <i>Alunos</i> |
| 1. Gerir utilizadores | 1.1. Tratar cadastro de utilizadores | 1.1.1. Inserir novo utilizador | | | |
| | | 1.1.2. Alterar utilizador | | | |
| | | 1.1.3. Excluir utilizador | | | |
| | | 1.1.4. Consultar utilizador | | | |
| 2. Gerir livros | 2.1. Tratar livros | 2.1.1. Inserir novo livro | | | |
| | | 2.1.2. Alterar livro | | | |
| | | 2.1.3. Excluir livro | | | |
| | | 2.1.4. Consultar livro | | | |

Figura 31 - Exemplo de parte de uma Matriz de Responsabilidades elaborada a partir de uma WBS

4.3.4 *Elaboração dos Diagramas de Casos de Uso iniciais*

A partir da WBS elaborado e de sua Matriz de Responsabilidades correspondente, deve-se elaborar os respectivos Diagramas de Casos de Uso. Para tanto, a estratégia geral sugerida é:

- a) transformar cada papel da Matriz de Responsabilidades em ator no caso de uso; e
- b) cada responsabilidade em uma atividade de seu caso de uso relacionado.

Quanto aos temas/assunto da Matriz de Responsabilidades, estes tornar-se-ão nos casos de uso de nível mais alto, os quais, naturalmente, serão compostos dos seus respectivos casos de uso detalhados. Por fim, as relações entre atores e atividades, nestes casos de uso, decorrerão da observação das responsabilidades de cada papel, conforme estabelecidas na Matriz de Responsabilidades. Do mesmo modo, as relações entre os casos de uso deverão ser analisadas se serão na forma de “*extends*” ou “*includes*”, conforme o contexto.

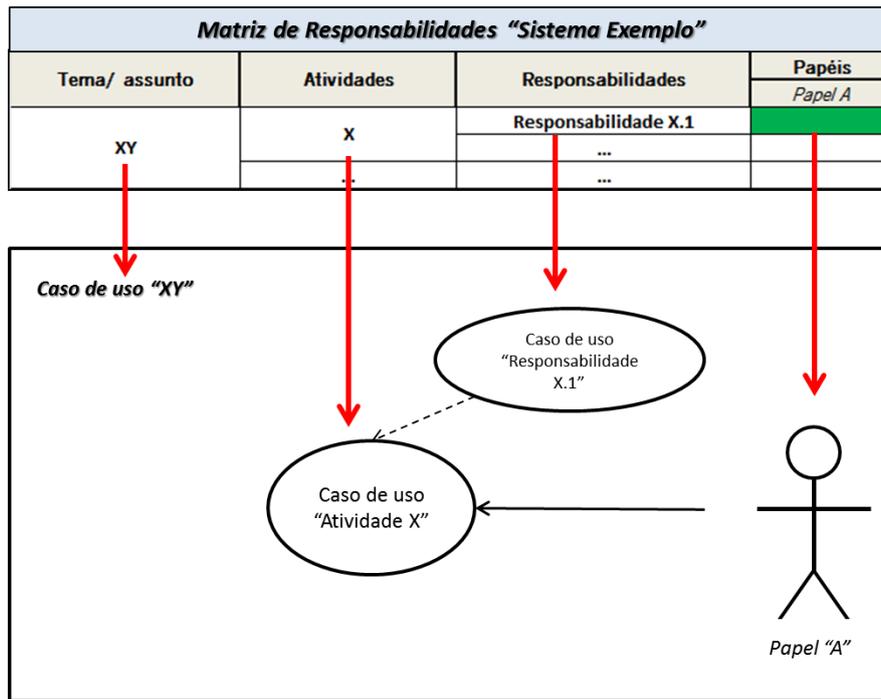


Figura 32 - Esquema de elaboração de casos de uso, a partir da Matriz de Responsabilidades

Por fim, para que se tenha uma percepção mais clara de como seria seu uso na prática, a Figura 33, abaixo, exemplifica como ficaria o caso de uso para o tema/assunto "1. Gerir utilizadores", citado na Matriz de Responsabilidades apresentada na Figura 31.

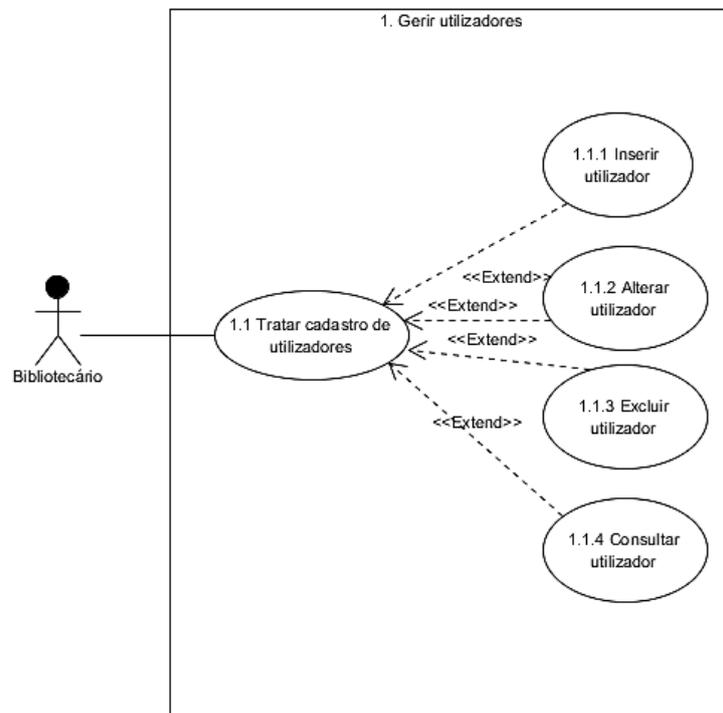


Figura 33 - Exemplo de caso de uso derivado de Matriz de Responsabilidades

4.3.5 Elaboração do Diagrama de Classes inicial

No que diz respeito ao Diagrama de Classes inicial, a ser confeccionado nessa etapa, este poderá ser elaborado tomando-se por base a Matriz de Responsabilidades feita em 4.3.3. Deste modo, como regra geral, sugere-se a abordagem recomendada por Larman [2008, p.165] para a descoberta das classes iniciais a serem incluídas na primeira versão deste artefato, ou seja, deve-se derivar uma única classe para cada membro da coluna “Papéis”, bem como uma para cada substantivo (com seu eventual qualificador) que apareça na coluna “Atividades” daquela matriz. Eventuais redundâncias encontradas também devem ser eliminadas. Assim, ao se proceder dessa forma, obter-se-á uma lista nominal de classes abrangente e sem redundâncias, a qual, por consequência, irá cobrir todos os requisitos funcionais.

Complementarmente, no que diz respeito aos métodos, estes também poderão ser derivados da Matriz de Responsabilidades. Nesse sentido, para obtê-los, recomenda-se que, para as classes derivadas da coluna “Papéis”, observe-se as relações entre os membros da coluna “Papéis” e a coluna “Atividades”, de forma que, onde houver relação entre ambas, deve-se transformar aquela “Atividade” num método daquele “Papel”. Por outro lado, quanto às classes derivadas de substantivos da coluna “Atividades”, deve-se transformar em métodos todas as atividades que façam referência àquele substantivo, conforme Figura 34, a seguir.

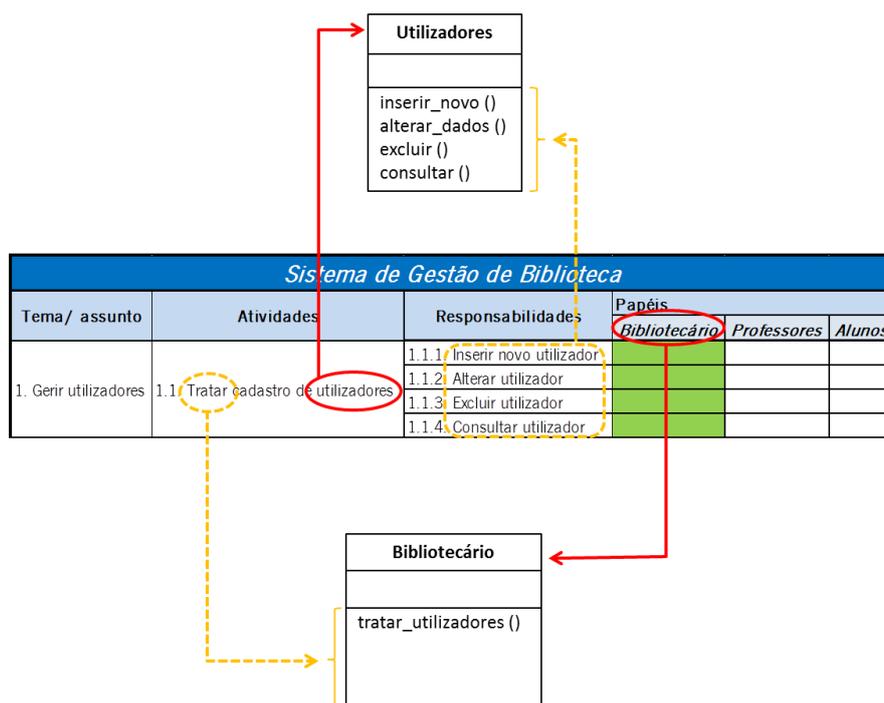


Figura 34 - Esquema de elaboração de classes e seus métodos a partir da Matriz de Responsabilidades

Por fim, as associações entre as classes inicialmente obtidas, também podem ser derivadas da própria matriz. Para tanto, deve-se observar que, onde houver relação entre papel e substantivos (das responsabilidades), tal deverá refletir-se numa associação entre as respectivas classes derivadas. Assim, ao observar-se a Figura 34, pode-se afirmar que as classes Bibliotecários e Utilizadores deverão estar associadas. A Figura 35, a seguir, demonstra essa associação.

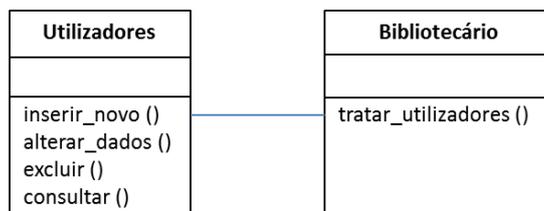


Figura 35 - Diagrama de Classes parcial derivado de uma Matriz de Responsabilidades

Cabe ressaltar, que trata-se, aqui, do Diagrama de Classes inicial, ou seja, este deverá ser complementado, sobretudo no que diz respeito aos seus atributos, pelas técnicas usualmente utilizadas para tanto por uma equipe responsável pelo desenvolvimento do sistema, tais como “aplicação de questionários, inspeção de documentos utilizados nos processos, *brainstorm*, *Joint Application Design* (JAD), entrevistas, observação *in loco*, entre outros” [Tonsig 2008, p.129].

4.4 Conclusão

Do que foi apresentado no presente capítulo, observa-se que, de fato, existe um conjunto de técnicas de Gestão de Projetos que podem influenciar a forma como alguns artefatos são construídos na fase de delineamento de requisitos em sistemas orientados a objetos. A possibilidade de conexão conceitual entre as dimensões Gestão de Projetos e Análise Orientada a Objetos, no que diz respeito ao delineamento de âmbito/requisitos, mostrou-se possível em termos conceituais, ainda que com a necessidade natural de algumas adequações nos documentos utilizados para este fim. A Tabela 4, a seguir, apresenta um resumo das referidas adequações, conforme propostas.

| Documentos de Gestão de Projetos | Resumo das adequações propostas |
|---------------------------------------|--|
| Termo de Abertura do Projeto | ▪ Eliminado da proposta de uso. |
| Declaração de Âmbito | ▪ Troca do enfoque de “gestão do projeto” para “requisitos do sistema”. ▪ Inclusão de item relativo aos tipos de utilizadores (atores). |
| <i>Work Breakdown Structure</i> (WBS) | ▪ Troca do enfoque de “gestão do projeto” para “requisitos do sistema”. ▪ Elaboração sob a ótica do cliente (requisitos funcionais). |
| Matriz de Responsabilidades | ▪ Incluídas as colunas “Atividades”, “Responsabilidades” e “Papéis” (Atores). |
| Método do Diagrama de Precedência | ▪ Eliminado da proposta de uso. |

Tabela 4 - Resumo das adequações propostas aos documentos de Gestão de Projetos

Tais ajustamentos, conforme explicado na introdução deste capítulo, decorreram do fato de que, tendo em vista que os contextos de utilização de documentos e artefatos são diferentes, percebia-se a necessidade de existir uma preocupação especial, por parte da equipe de desenvolvimento de sistemas, em ajustar seu enfoque, quando do uso de técnicas documentais emprestadas da Gestão de Projetos. Nesse sentido, tal equipe deverá concentrar-se em aplicá-las abstraindo-se do foco em gestão do processo de desenvolvimento do sistema, voltando suas atenções somente para os requisitos funcionais que atendam às necessidades do cliente.

De todo o modo, a similaridade de finalidade entre alguns documentos/artefatos das fases iniciais daquelas duas dimensões mostrou-se particularmente evidente quanto a três questões cruciais: a) o quê; b) quem; e c) como. Sem dúvida, ao tentarem responder a tais questões, tanto documentos, em Gestão de Projetos, quanto artefatos, em Análise Orientada a Objetos, acabam por ficar com objetivos semelhantes, razão pela qual ficou viável um intercâmbio de aplicabilidade.

A partir da verificação desta realidade, tornou-se possível propor um processo de conexão entre as ditas dimensões, de forma a se concretizar o uso efetivo de documentos de Gestão de Projetos em desenvolvimento de sistemas orientados a objetos. Nesse sentido, apesar da impossibilidade de uso do Termo de Abertura de Projeto e do Diagrama de Precedência, conforme explicado no item 4.2, outros se mostraram bastante viáveis, ainda que com pequenos ajustes, como foi o caso da Declaração de Âmbito, da WBS e da Matriz de Responsabilidades.

De fato, os esquemas de transposição de uso dessas técnicas de Gestão de Projetos para Análise Orientada a Objetos, conforme ilustrados pelas figuras 30 (Matriz de Responsabilidades, a partir da WBS e da Declaração de Âmbito), 32 (Casos de Uso, a partir da Matriz de Responsabilidades) e 34 (Diagrama de Classes, a partir da Matriz de Responsabilidades), mostram que a execução de tais procedimentos pode ser feita de maneira sistemática, pelo que, não por acaso, foram apresentadas dentro do corpo de uma proposta de processo de conexão, detalhada no tópico 4.3. Neste sentido, a simplicidade e consistência dos procedimentos apresentados permitem pressupor que sua aplicação, num caso prático, poderá conduzir a resultados satisfatórios.

5. Estudo de Caso

5.1 Considerações iniciais

O presente estudo de caso concentra-se sobre um sistema que pretendia atender às necessidades de informação sobre métricas de qualidade dos serviços prestados pela Santa Casa de Misericórdia de Guimarães (SCMG). Todavia, segundo informações do orientador desta dissertação, o sistema em causa não chegou a ser efetivamente implantado. De todo o modo, o Relatório Final de Desenvolvimento [Lucas, et al., 2011], o qual continha toda a documentação relacionada ao sistema a ser apresentado, foi disponibilizado pela equipe responsável pelo seu desenvolvimento e implantação.

Assim, apesar de o referido relatório conter mais de 20 (vinte) documentos acerca do sistema, o ponto de partida deste estudo de caso baseou-se exclusivamente na descrição textual contida em apenas um único artefato, nomeadamente, “Visão”, cuja finalidade era “apresenta[r] as principais necessidades do cliente e a forma de como o [...] produto irá colmatar as mesmas”, bem como de “explicitar o produto desenvolvido (software) e as suas principais características” [Lucas, et al. 2011, p. 5]. Por conseguinte, todos os diagramas relacionados à análise do sistema (Casos de Uso, Classes, Atividades, Sequência, etc.) foram integralmente desconsiderados para fins de levantamento dos requisitos deste estudo.

De todo modo, após a aplicação da proposta de conexão conforme contida no tópico 4.3 ao presente estudo de caso, foi possível uma comparação entre os documentos originais de desenvolvimento do sistema e aqueles produzidos segundo a abordagem sugerida.

5.2 Sobre a instituição e o sistema SIAQ-SCMG

A Santa Casa de Misericórdia de Guimarães (SCMG) é uma Instituição Privada de Solidariedade Social (IPSS) que presta uma ampla gama de serviços e sob a qual encontram-se: o Lar Rainha Dona Leonor, a Casa de Repouso João Antunes Guimarães (Donim), o Centro de Solidariedade Humana Professor Emídio Guerreiro, o Lar de S. Paio, a Residência das Trinas, o Centro de Atividades Ocupacionais, o Lar Residencial Alecrim e uma Unidade de Cuidados Continuados de Longa Duração e Manutenção. No seu antigo Hospital estão implantados vários serviços de saúde, entre os quais um serviço de Endoscopia Digestiva e Colonoscopia, uma unidade de Hemodiálise, um Centro de Medicina Nuclear, um serviço de Análises Clínicas, uma clínica de Imagiologia e outra de Fisiatria [SCMG, 2012].

Relativamente a sua estruturação organizacional, a SCMG é administrada por uma Assembleia Geral, a qual estão subordinadas dois diretores, bem como duas assessorias, a saber: Diretor Coordenador, Diretor de Qualidade, Assessoria Jurídica e Assessoria Económico-Financeira. A Figura 36, a seguir, apresenta o organograma da instituição, destacando-se, em amarelo-claro, os setores que seriam abrangidos pelo sistema pretendido.

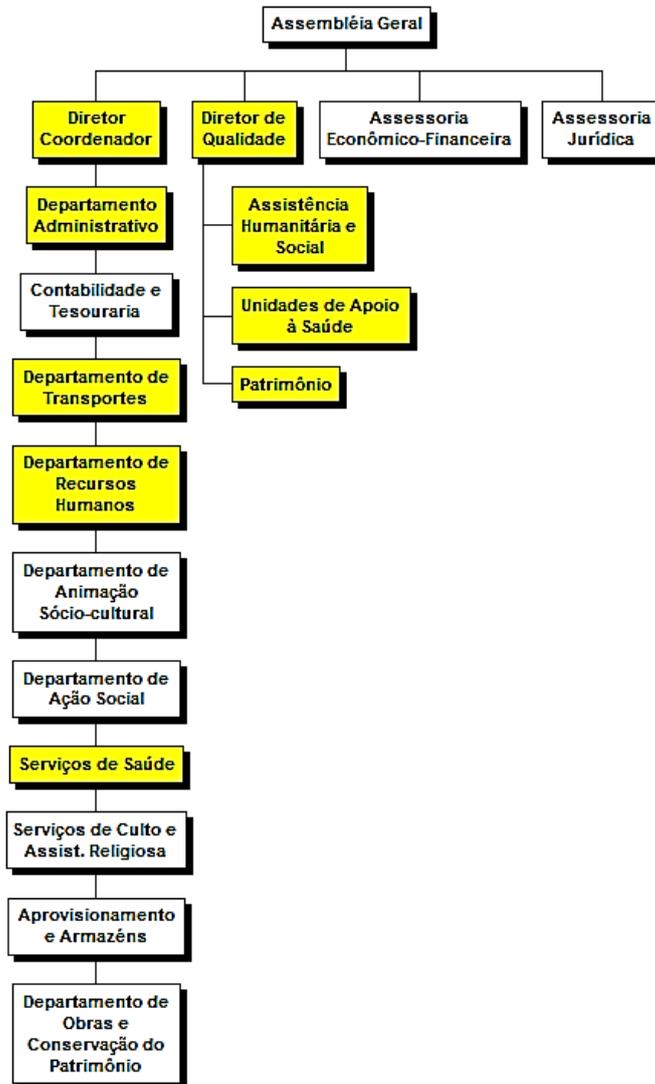


Figura 36 - Organograma da Santa Casa de Misericórdia de Guimarães (SCMG) [SCMG, 2012]

A principal motivação para a implantação de um sistema informático, na Santa Casa de Misericórdia, estava relacionada à necessidade de se recolherem dados para melhorar o nível de qualidade ISO-9001, pela qual a instituição já era certificada. Segundo a documentação obtida, foi constatado que não existia uma forma simples e eficaz que permitisse acessar os dados recolhidos pela organização para obter métricas que auxiliassem a melhoria do sistema de qualidade implementado. A

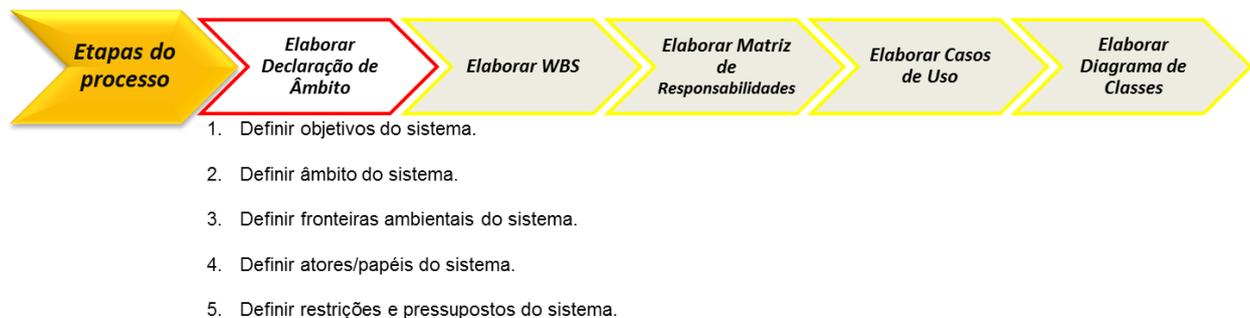
forma encontrada pela SCMG para contornar este problema foi a utilização de documentos impressos que deveriam ser preenchidos e entregues à direção de cada unidade, de modo que os responsáveis pelas mesmas, ao seu turno, deveriam passá-los para o formato digital, em documentos de *MS Excel* ou *MS Word*, e entregá-los à Direção de Qualidade. Esta forma de recolha de métricas carecia de eficiência e agilidade. Assim, o principal objetivo do desenvolvimento desse sistema informático foi o de facilitar a obtenção e elaboração de estatísticas para a Direção de Qualidade da SCMG [Lucas, et al., 2011].

5.3 Aplicação do processo proposto



Conforme explicado na abertura deste capítulo, para que este estudo não sofresse influências dos casos de uso contidos na documentação obtida na fase anterior, tais não foram considerados para fins de levantamento dos requisitos funcionais deste estudo de caso. Assim, para se cumprirem as atividades da presente fase, foram seguidas as etapas do processo proposto, conforme definidos no capítulo 4.

5.3.1 Elaboração da Declaração de Âmbito



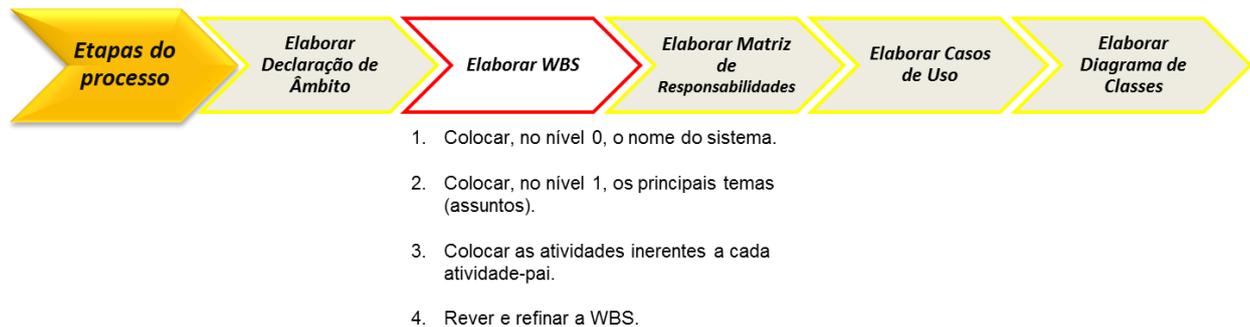
Seguindo a rotina sugerida no tópico 4.3, a Declaração de Âmbito é o primeiro documento a ser elaborado. Desta forma, ao atender à estrutura ali definida, aplicando-a sobre o presente estudo de caso, obteve-se o seguinte:

| Declaração de Âmbito | |
|--|--|
| Sistema de Apoio à Qualidade – Santa Casa de Misericórdia de Guimarães (SIAQ-SCMG) | |
| Objetivos do sistema | Recolher métricas para aferir a qualidade dos processos executados. |
| Âmbito do sistema | Para atingir o principal objetivo do sistema faz-se necessária a criação dos módulos a seguir, conforme especificação do cliente: Autenticar o sistema, Gerir utentes, Gerir espaços, Gerir atendimentos e consultas médicas, Gerir apoio domiciliário, Gerir meios, Gerir funcionários, Gerir métricas e Gerir relatórios. |
| Fronteiras ambientais do sistema | O Sistema de Apoio à Qualidade será um sistema de plataforma WEB e deverá possuir uma interface com características <i>User Friendly</i> . |
| Atores/papéis | <p>Relativamente aos atores/papéis que irão interagir com o sistema, listam-se os seguintes:</p> <ul style="list-style-type: none"> ▪ Provedor: Responsável por supervisionar o projeto, decidir admissão de utentes, orçamentos por área, gestão de recursos humanos e aprovação de Investimentos. ▪ Diretor de qualidade: Responsável por gerir o cumprimento por parte dos funcionários da certificação ISO 9001:2008. ▪ Diretor de Unidade: Responsável por zelar pelo bem estar dos utentes e garantir a higienização dos mesmos. ▪ Médico: Responsável por prestar serviços na área de saúde nos lares e apoio domiciliário. ▪ Enfermeira: Responsável por prestar serviços na área de saúde nos lares e apoio domiciliário. ▪ Administrativo: Responsável por atribuir tarefas aos médicos, enfermeiros e auxiliares. ▪ Auxiliar de saúde: Responsável por prestar serviços aos utentes. |
| Restrições e pressupostos do sistema | A linguagem de programação a ser utilizada para implementação será Java, Javascript e PHP. O método a ser utilizado para desenvolver a arquitetura do <i>Software</i> é o <i>Four Step Rule Set</i> – 4SRS. Atualmente a Santa Casa de Misericórdia de Guimarães é certificada com a Norma de qualidade ISO 9001/2008. Outras restrições tecnológicas do cliente: a) Sistema operacional: <i>Microsoft Windows</i> XP ou <i>Windows 7</i> ; b) Navegador: <i>Internet Explorer 9</i> , <i>Google Chrome 10</i> ou <i>Mozilla Firefox 4</i> . |

Tabela 5 - Declaração de âmbito sugerida para o Estudo de Caso

Conforme pode-se observar, buscou-se levantar os principais atores/papéis do SIAQ-SCMG, bem como suas respectivas responsabilidades, a partir do documento “Visão” [Lucas, et al., 2011]. Em que pese o sistema controlar a aplicação dos diversos serviços de saúde aos utentes, estes, segundo a documentação disponibilizada, não interagem diretamente com o sistema. Desta forma, não podem ser conceitualmente entendidos como atores.

5.3.2 Elaboração do WBS



A partir do documento “Visão” [idem], buscou-se levantar os principais requisitos funcionais do SIAQ-SCMG. Mais uma vez, tendo-se em conta os cuidados explicados na fase anterior, estes foram elaborados a partir do item “Características do produto” contido naquele mesmo documento. Todavia, é importante ressaltar que os requisitos listados a seguir foram concebidos segundo o entendimento desta autora relativamente às informações ali contidas.

Desta forma, conforme será apresentado e discutido no item 5.4, à frente, os requisitos da referida documentação e os que seguem apresentam diferenças notáveis, mas, naturalmente, também guardam muitas similaridades. De todo o modo, os seguintes requisitos funcionais foram elencados:

| Requisitos funcionais do SIAQ - SCMG | | |
|--------------------------------------|--------------------------------------|---|
| 1. Gerir utilizadores | 1.1. Tratar cadastro de utilizadores | 1.1.1. Inserir novo utilizador |
| | | 1.1.2. Alterar utilizador |
| | | 1.1.3. Excluir utilizador |
| | | 1.1.4. Consultar utilizador |
| 2. Gerir utentes | 2.1. Tratar formulários | 2.1.1. Liberar acesso às unidades |
| | | 2.1.2. Registrar pedido de apoio domiciliário |
| | 2.2. Tratar cadastro de utentes | 2.2.1. Inserir novo utente |
| | | 2.2.2. Alterar dados de utentes |
| | | 2.2.3. Excluir utente |
| | | |

| | | |
|--|--|---|
| | | 2.2.4. Consultar fichas de utentes |
| | 2.3. Tratar ementas | 2.3.1. Elaborar ementa |
| | | 2.3.2. Alterar ementa |
| | | 2.3.3. Excluir ementa |
| | | 2.3.4. Consultar ementa |
| | 2.4. Tratar cuidados regulares | 2.4.1. Inserir cuidados regulares |
| | | 2.4.2. Alterar cuidados regulares |
| | | 2.4.3. Excluir cuidados regulares |
| | | 2.4.4. Consultar cuidados regulares |
| | 2.5. Tratar cuidados suplementares | 2.5.1. Inserir cuidados suplementares |
| | | 2.5.2. Alterar cuidados suplementares |
| | | 2.5.3. Excluir cuidados suplementares |
| | | 2.5.4. Consultar cuidados suplementares |
| 3. Gerir espaços | 3.1. Tratar espaços da unidade | 3.1.1. Definir espaços |
| | | 3.1.2. Editar espaços |
| | 3.2. Definir horários de limpeza aos espaços | 3.2.1. Calendarizar mudas de camas |
| | | 3.2.2. Calendarizar limpezas aos sanitários |
| | | 3.2.3. Calendarizar limpezas aos espaços |
| | 3.3. Tratar registros de limpeza aos espaços | 3.3.1. Inserir registo de limpeza |
| | | 3.3.2. Alterar registo de limpeza |
| | | 3.3.3. Excluir registo de limpeza |
| | | 3.3.4. Consultar registo de limpeza |
| | 3.4. Tratar espaços no sistema | 3.4.1. Inserir espaços da unidade |
| | | 3.4.2. Editar espaços da unidade |
| | | 3.4.3. Excluir espaços da unidade |
| | | 3.4.4. Consultar espaços da unidade |
| 4. Gerir Atendimentos e consultas médicas | 4.1. Tratar consultas | 4.1.1. Agendar consulta |
| | | 4.1.2. Alterar consulta |
| | | 4.1.3. Excluir consulta |
| | | 4.1.4. Consultar agenda de consultas |
| | 4.2. Tratar tratamentos | 4.2.1. Agendar tratamento |
| | | 4.2.2. Alterar tratamento |
| | | 4.2.3. Excluir tratamento |
| | | 4.2.4. Consultar agenda de tratamentos |
| | 4.3. Tratar cuidados | 4.3.1. Registrar cuidados especiais |
| | | 4.3.2. Alterar cuidados especiais |

| | | | | |
|------------------------------------|--------------------------------------|-------------------------------------|-------------------------------|----------------------------------|
| | | 4.3.3. Excluir cuidados especiais | | |
| | | 4.3.4. Consultar cuidados especiais | | |
| 5. Gerir apoio domiciliário | 5.1. Tratar avaliação de candidatura | 5.1.1. Inserir candidatura | | |
| | | 5.1.2. Registrar decisão | | |
| | 5.2. Tratar serviços | 5.2.1. Agendar serviços | | |
| | | 5.2.2. Alterar serviços | | |
| | | 5.2.3. Comunicar serviços | | |
| | | 5.2.4. Excluir serviços | | |
| | | 5.2.5. Suspender serviços | | |
| | | 5.2.6. Consultar serviços suspensos | | |
| | 5.3. Tratar serviços prestados | 5.3.1. Registrar serviços prestados | | |
| | | 5.3.2. Alterar serviços prestados | | |
| | | 5.3.3. Excluir serviços prestados | | |
| | | 5.3.4. Consultar serviços prestados | | |
| 6. Gerir meios | 6.1. Tratar viaturas | 6.1.1. Inserir viatura | | |
| | | 6.1.2. Alterar viatura | | |
| | | 6.1.3. Excluir viatura | | |
| | | 6.1.4. Consultar viatura | | |
| | 6.2. Tratar equipamentos | 6.2.1. Inserir equipamento | | |
| | | 6.2.2. Alterar equipamento | | |
| | | 6.2.3. Excluir equipamento | | |
| | | 6.2.4. Consultar equipamento | | |
| | | 7. Gerir funcionário | 7.1. Tratar escala de serviço | 7.1.1. Inserir escala de serviço |
| | | | | 7.1.2. Alterar escala de serviço |
| 7.1.3. Excluir escala de serviço | | | | |
| 7.1.4. Consultar escala de serviço | | | | |
| 7.2. Tratar funcionários | 7.2.1. Inserir dados | | | |
| | 7.2.2. Alterar dados | | | |
| | 7.2.3. Inativar dados | | | |
| | 7.2.4. Consultar funcionário | | | |
| | 7.2.5. Eliminar funcionário | | | |
| 8. Gerir métricas | 8.1. Tratar métricas do utente | 8.1.1. Consultar | | |
| | | 8.1.2. Imprimir | | |
| | | 8.1.3. Exportar para ficheiro Excel | | |
| | 8.2. Tratar métricas do espaço | 8.2.1. Consultar | | |
| | | 8.2.2. Imprimir | | |

| | | |
|-------------------|--|-------------------------------------|
| | | 8.2.3. Exportar para ficheiro Excel |
| | 8.3. Tratar métricas de apoio domiciliar | 8.3.1. Consultar |
| | | 8.3.2. Imprimir |
| | | 8.3.3. Exportar para ficheiro Excel |
| 9. Gerir | 9.1. Tratar relatório | 9.1.1. consultar relatório |
| relatórios | | 9.1.2. Selecionar relatório |
| | | 9.1.3. Imprimir |

Tabela 6 - Requisitos funcionais do SIAQ – SCMG

Após a identificação dos requisitos funcionais a partir do documento “Visão” citado, estes foram organizados e agrupados em temas afins, de forma a comporem os diversos ramos da WBS, conforme sugerido na técnica proposta. Desta forma, obteve-se a WBS contida na Figura 37, a qual, conforme se observa, apresenta uma visão do sistema como um todo, o que facilita, principalmente por parte do cliente, a compreensão do que o sistema em questão deverá fazer. Do mesmo modo, eventuais inserções, exclusões e alterações de requisitos, ainda na fase de levantamento, tornam-se atividades triviais.

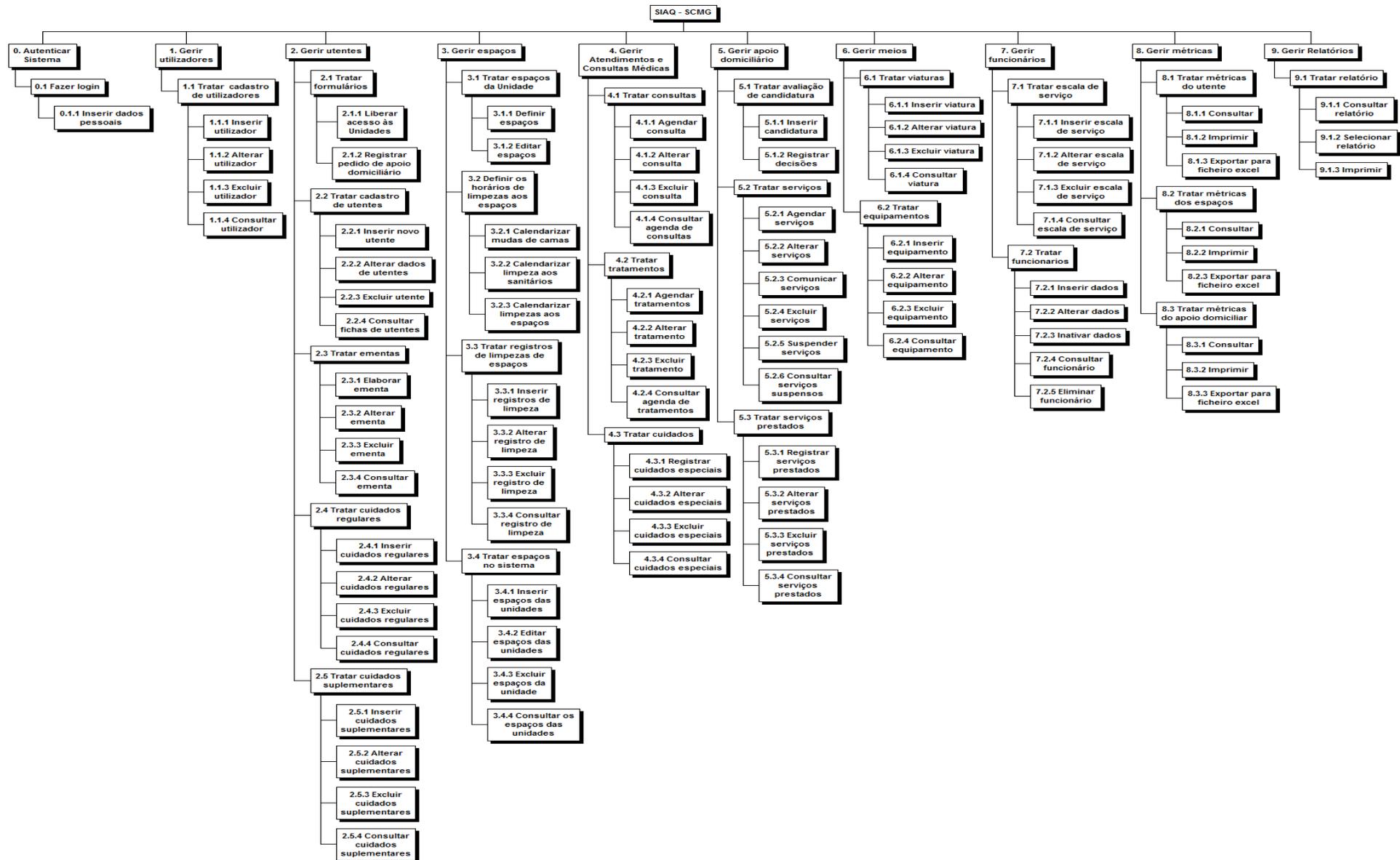
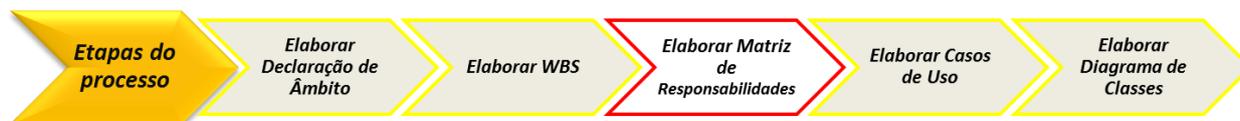


Figura 37 - WBS elaborada a partir dos requisitos funcionais do SIAQ - SCMG

5.3.3 *Elaboração da Matriz de Responsabilidades*



1. Transpor toda a estrutura da WBS elaborada na etapa anterior, com todos os seus níveis e subníveis, para a matriz de responsabilidades.
2. O nível 1 deverá ser incluído numa coluna chamada "Tema/Assunto"; o nível 2, numa chamada "Atividades"; por fim, o nível 3, numa coluna chamada "Responsabilidades".
3. Definir os atores, obtidos na Declaração de Âmbito, como "papéis", em colunas próprias.
4. Para cada responsabilidade, assinalar a interação com cada papel, se houver.

A partir da identificação dos principais atores, conforme citados na Declaração de Âmbito sugerida, bem como dos requisitos funcionais levantados, foi possível elaborar a Matriz de Responsabilidades que segue. Todavia, devido ao elevado número de atividades e tarefas correlatas, a matriz constante da Tabela 7 apresenta somente os 5 (cinco) primeiros temas (assuntos) a serem geridos por meio do sistema SIAQ-SCMG.

Deste modo, apresenta-se, a seguir, a visão parcial da referida matriz. A visão completa da mesma pode ser encontrada no Anexo 4.

| SIAQ - SCMG | | | | | | | | | |
|---|--|---|--------------|----------------|----------------|----------|--------|------------|---------------|
| Tema/ assunto | Atividades | Responsabilidades | Papéis | | | | | | |
| | | | Dir. Unidade | Dir. Qualidade | Administrativo | Provedor | Médico | Enfermeira | Aux. de saúde |
| 0. Autenticar sistema | 0.1. Fazer login | 0.1.1. Inserir dados pessoais | | | | | | | |
| 1. Gerir utilizadores | 1.1. Tratar cadastro de utilizadores | 1.1.1. Inserir novo utilizador | | | | | | | |
| | | 1.1.2. Alterar utilizador | | | | | | | |
| | | 1.1.3. Excluir utilizador | | | | | | | |
| | | 1.1.4. Consultar utilizador | | | | | | | |
| 2. Gerir utentes | 2.1. Tratar formulários | 2.1.1. Liberar acesso às unidades | | | | | | | |
| | | 2.1.2. Registrar pedido de apoio domiciliário | | | | | | | |
| | 2.2. Tratar cadastro de utentes | 2.2.1. Inserir novo utente | | | | | | | |
| | | 2.2.2. Alterar dados de utentes | | | | | | | |
| | | 2.2.3. Excluir utente | | | | | | | |
| | | 2.2.4. Consultar fichas de utentes | | | | | | | |
| | 2.3. Tratar ementas | 2.3.1. Elaborar ementa | | | | | | | |
| | | 2.3.2. Alterar ementa | | | | | | | |
| | | 2.3.3. Excluir ementa | | | | | | | |
| | | 2.3.4. Consultar ementa | | | | | | | |
| | 2.4. Tratar cuidados regulares | 2.4.1. Inserir cuidados regulares | | | | | | | |
| | | 2.4.2. Alterar cuidados regulares | | | | | | | |
| | | 2.4.3. Excluir cuidados regulares | | | | | | | |
| | | 2.4.4. Consultar cuidados regulares | | | | | | | |
| | 2.5. Tratar cuidados suplementares | 2.5.1. Inserir cuidados suplementares | | | | | | | |
| | | 2.5.2. Alterar cuidados suplementares | | | | | | | |
| 2.5.3. Excluir cuidados suplementares | | | | | | | | | |
| 2.5.4. Consultar cuidados suplementares | | | | | | | | | |
| 3. Gerir espaços | 3.1. Tratar espaços da unidade | 3.1.1. Definir espaços | | | | | | | |
| | | 3.1.2. Editar espaços | | | | | | | |
| | 3.2. Definir horários de limpeza aos espaços | 3.2.1. Calendarizar mudas de camas | | | | | | | |
| | | 3.2.2. Calendarizar limpezas aos sanitários | | | | | | | |
| | | 3.2.3. Calendarizar limpezas aos espaços | | | | | | | |
| | 3.3. Tratar registros de limpeza aos espaços | 3.3.1. Inserir registro de limpeza | | | | | | | |
| | | 3.3.2. Alterar registro de limpeza | | | | | | | |
| | | 3.3.3. Excluir registro de limpeza | | | | | | | |
| | | 3.3.4. Consultar registro de limpeza | | | | | | | |
| | 3.4. Tratar espaços no sistema | 3.4.1. Inserir espaços da unidade | | | | | | | |
| | | 3.4.2. Editar espaços da unidade | | | | | | | |
| | | 3.4.3. Excluir espaços da unidade | | | | | | | |
| 3.4.4. Consultar espaços da unidade | | | | | | | | | |
| 4. Gerir Atendimentos e consultas médicas | 4.1. Tratar consultas | 4.1.1. Agendar consulta | | | | | | | |
| | | 4.1.2. Alterar consulta | | | | | | | |
| | | 4.1.3. Excluir consulta | | | | | | | |
| | | 4.1.4. Consultar agenda de consultas | | | | | | | |
| | 4.2. Tratar tratamentos | 4.2.1. Agendar tratamento | | | | | | | |
| | | 4.2.2. Alterar tratamento | | | | | | | |
| | | 4.2.3. Excluir tratamento | | | | | | | |
| | | 4.2.4. Consultar agenda de tratamentos | | | | | | | |
| | 4.3. Tratar cuidados | 4.3.1. Registrar cuidados especiais | | | | | | | |
| | | 4.3.2. Alterar cuidados especiais | | | | | | | |
| | | 4.3.3. Excluir cuidados especiais | | | | | | | |
| | | 4.3.4. Consultar cuidados especiais | | | | | | | |

Tabela 7 - Visão parcial da matriz de responsabilidades SIAQ - SCMG

5.3.4 *Elaboração dos Casos de Uso iniciais*



1. Transformar cada papel da Matriz de Responsabilidades em ator do caso de uso.
2. Transformar cada responsabilidade em atividade do seu respectivo caso de uso.
3. Estabelecer as relações entre atores e atividades conforme a referida Matriz.

A partir da WBS e da Matriz de Responsabilidades obtidas, foram elaborados os casos de uso iniciais do sistema em questão. No entanto, para fins deste estudo de caso e simplificação da análise comparativa a ser feita posteriormente, estes não foram elaborados na sua totalidade, pois que o objetivo do presente trabalho não é por em causa a documentação do SIAQ-SCMG existente, mas, tão somente, verificar a aplicabilidade da abordagem proposta.

Neste sentido, então, para os efeitos desta etapa, foram considerados exclusivamente os requisitos relativos ao tema/assunto “2. Gerir utentes”.

▪ **Tema/assunto “2.Gerir utentes”**

| Tema/ assunto | Atividades | Responsabilidades | Papéis | | | | | | |
|------------------|------------------------------------|---|--------------|----------------|----------------|----------|--------|------------|---------------|
| | | | Dir. Unidade | Dir. Qualidade | Administrativo | Provedor | Médico | Enfermeira | Aux. de saúde |
| 2. Gerir utentes | 2.1. Tratar formulários | 2.1.1. Liberar acesso às unidades | | | | | | | |
| | | 2.1.2. Registrar pedido de apoio domiciliário | | | | | | | |
| | 2.2. Tratar cadastro de utentes | 2.2.1. Inserir novo utente | | | | | | | |
| | | 2.2.2. Alterar dados de utentes | | | | | | | |
| | | 2.2.3. Excluir utente | | | | | | | |
| | | 2.2.4. Consultar fichas de utentes | | | | | | | |
| | 2.3. Tratar ementas | 2.3.1. Elaborar ementa | | | | | | | |
| | | 2.3.2. Alterar ementa | | | | | | | |
| | | 2.3.3. Excluir ementa | | | | | | | |
| | | 2.3.4. Consultar ementa | | | | | | | |
| | 2.4. Tratar cuidados regulares | 2.4.1. Inserir cuidados regulares | | | | | | | |
| | | 2.4.2. Alterar cuidados regulares | | | | | | | |
| | | 2.4.3. Excluir cuidados regulares | | | | | | | |
| | | 2.4.4. Consultar cuidados regulares | | | | | | | |
| | 2.5. Tratar cuidados suplementares | 2.5.1. Inserir cuidados suplementares | | | | | | | |
| | | 2.5.2. Alterar cuidados suplementares | | | | | | | |
| | | 2.5.3. Excluir cuidados suplementares | | | | | | | |
| | | 2.5.4. Consultar cuidados suplementares | | | | | | | |

Tabela 8 - Visão parcial da matriz de responsabilidades – Tema/assunto " 2. Gerir utentes"

Conforme pode-se observar pela tabela acima, o Diretor de Unidade interage com as atividades: “2.2 Tratar cadastro de utentes” e “2.3 Tratar ementas”. O Diretor de Qualidade interage somente com a atividade “2.3 Tratar ementas”. O Administrativo interage com as atividades “2.1 Tratar formulários”, “2.2 Tratar cadastro de utentes” e “2.3 Tratar ementas”. Os papéis de Médico e a Enfermeira interagem com as atividades “2.2 Tratar cadastro de utentes”, “2.3 Tratar ementas”, “2.4 Tratar cuidados regulares” e

“2.5 Tratar cuidados suplementares”. Por fim, o Auxiliar de Saúde interage com as atividades “2.4 Tratar cuidados regulares” e “2.5 Tratar cuidados suplementares”.

Deste modo, ao aplicarem-se as orientações contidas no tópico 4.3 ao estudo de caso em apreço, obtém-se o caso de uso apresentado a seguir, na Figura 38.

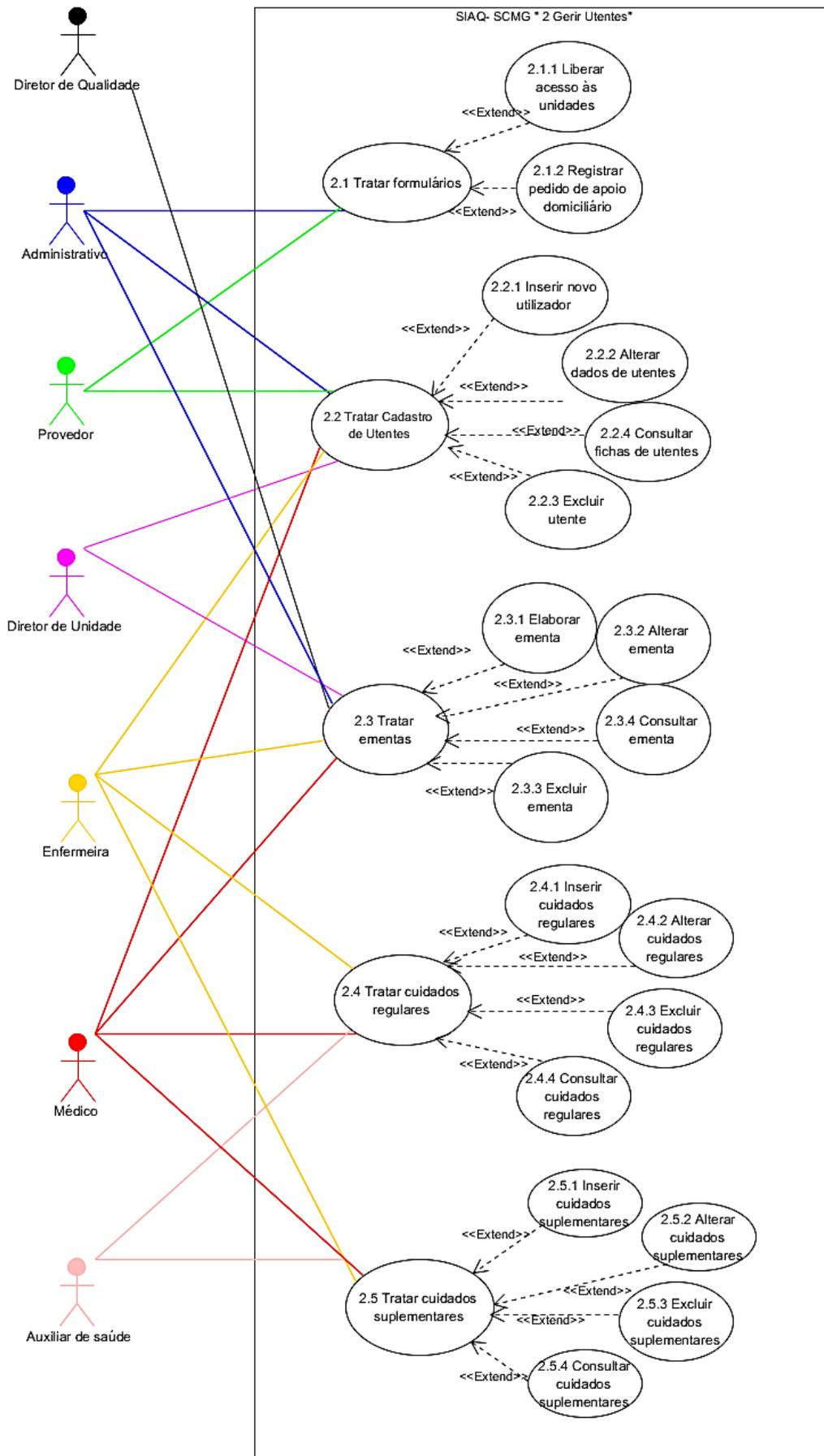


Figura 38 - Caso de uso "Gerir utentes"

5.3.5 *Elaboração do Diagrama de Classes inicial*



Para a execução dessa etapa, utilizou-se a abordagem proposta no tópico 4.3 do presente trabalho. Desta forma, ao se levantar os substantivos (com seus eventuais qualificadores) existentes em cada atividade da WBS elaborada em 5.3.2 (Figura 37), foram identificadas as seguintes classes, na atividade "2 - Gerir utentes", que irão compor o Diagrama de Classes inicial:

- Formulários
- Utentes
- Ementas
- Cuidados regulares
- Cuidados suplementares
- Unidades
- Apoios domiciliários

Em complemento a estas classes, ainda seguindo abordagem sugerida no tópico 4.3 deste trabalho, foram buscados os papéis constantes da Matriz de Responsabilidades, conforme consta na Tabela 7 em 5.3.3. Desta forma, então, a partir das atividades em "2 – Gerir utentes", foram obtidas as seguintes classes adicionais:

- Diretor de Qualidade
- Diretor de Unidade
- Administrativo
- Auxiliar de Saúde
- Enfermeiro
- Médico
- Provedor

Para fins de levantamento dos métodos que compuseram as respectivas classes iniciais, buscou-se seguir as orientações contidas no tópico 4.3.5. Assim, as referidas classes ficaram da seguinte forma:

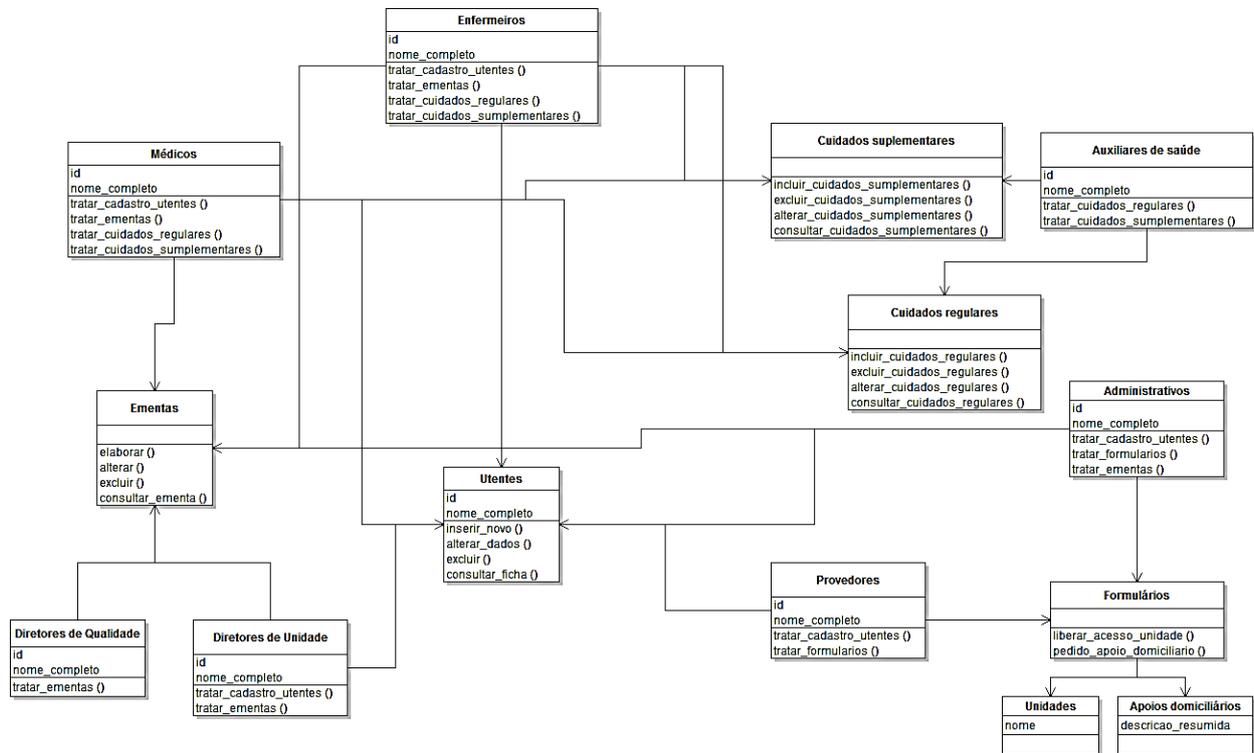


Figura 39 - Diagrama de Classes inicial para "2. Gerir Utentes"

Note-se que, conforme previamente mencionado no tópico 4.3.5, a figura acima refere-se apenas a uma versão inicial do Diagrama de Classes. Assim, a partir desta, a equipe responsável pelo desenvolvimento deverá aplicar as técnicas de refinamento de requisitos as quais esteja habituada, de forma a complementá-lo, particularmente no que diz respeito aos atributos daquelas classes.

5.4 Considerações sobre o processo

Ao se aplicar a proposta de conexão ao presente Estudo de Caso, conforme recomendada no tópico 4.3, observou-se que, em relação à documentação original, os novos documentos e artefatos produzidos mostraram-se, de fato, bastante coerentes e coesos. Efetivamente, conforme se pode observar ao longo deste capítulo, o uso intensivo de técnicas como Declaração de Âmbito, WBS e Matriz de Responsabilidades, forneceu inúmeros *inputs* relevantes para diversos artefatos típicos num desenvolvimento orientado a objetos, nomeadamente os Diagramas de Casos de Uso e Diagrama de Classes.

De todo modo, a fim de viabilizar uma comparação mais clara entre o delineamento de requisitos existente na documentação original, em contraposição ao seu equivalente derivado da presente proposta, foi elaborada uma WBS completa para aquela documentação (Anexo 1), bem como outra para o presente Estudo de Caso (Anexo 2). Assim sendo, algumas diferenças tornaram-se evidentes entre as duas abordagens.

De imediato, ao compararem-se as respectivas WBS contidas nos Anexos citados, constata-se uma diferença significativa no número de requisitos levantados, pois, abstraindo-se as atividades de nível 0 e 1 (respectivamente, nome do sistema e temas/assuntos abrangidos), verifica-se que a documentação original apresenta um total de 68 (sessenta e oito) requisitos funcionais listados [Lucas, et al., 2011, p.12], ao passo que, após aplicarem-se as técnicas sugeridas em 4.3, esse total elevou-se a 113 (cento e treze) requisitos de mesma natureza, ou seja, uma expressiva diferença de 45 (quarenta e cinco) requisitos funcionais adicionados. De fato, observa-se que não foram incluídas, na documentação original, atividades funcionais importantes como: excluir e consultar utilizador; inserir e excluir utentes; alterar, excluir e consultar cuidados regulares; e alterar e excluir consultas médicas, para citarem-se apenas alguns.

De modo semelhante, foram observadas diferenças no número e distribuição de atividades de nível 1 da WBS de cada abordagem (original e proposta), as quais, vale lembrar, correspondem aos temas/assuntos tratados pelo sistema em questão. Nesse sentido, nota-se que a abordagem da documentação original continha um total de 8 (oito) módulos funcionais, em oposição aos 10 (dez) percebidos por meio da técnica proposta. Para além disso, é possível notar que determinadas atividades estão estruturadas de modo diferente na WBS relativa à documentação original, como, por exemplo, a atividade “Disponibilizar impressos”, a qual encontra-se no nível 1 daquela WBS, porém com apenas uma única subatividade (“Selecionar”), o que pode ser uma evidência de falha na compreensão dessa atividade e/ou de sua relação com outras atividades que possam lhe dizer respeito. Não por acaso, na nova WBS, essa mesma funcionalidade foi transferida para dentro do tema/assunto “9. Gerir relatórios”, o qual possui três subatividades (vide Anexo 2).

Relativamente à Matriz de Responsabilidades, algumas comparações também podem ser feitas a partir desse documento. Para tanto, além da matriz já produzida para este Estudo de Caso, elaborou-se uma outra a partir da concepção original do sistema SIAQ-SCMG, conforme estabelecida em sua própria documentação. As matrizes completas, tanto do entendimento original, quanto do decorrente da aplicação da proposta, encontram-se disponíveis nos Anexos Anexo 3 e Anexo 4, respectivamente. No que diz

respeito às colunas “Tema/assunto”, “Atividades” e “Responsabilidades” pouco há a acrescentar, visto que o conteúdo destas reflete, necessariamente, suas respectivas WBS, as quais já foram discutidas nos parágrafos anteriores.

Por outro lado, há que se fazer notar alguns pontos acerca da coluna “Papéis” e de sua relação com a coluna “Responsabilidades”. Inicialmente, conforme se pode observar pelo Anexo 3, o qual apresenta a matriz relacionada à concepção original do sistema, observa-se que diversos “papéis” (ou atores, conforme o contexto) podem efetuar todas as operações previstas no sistema SIAQ-SCMG, nomeadamente, Diretor de Unidade, Diretor de Qualidade e Provedor. Tendo-se em conta que estes três cargos são de nível de Diretoria (ou *board*), parece bastante improvável que as pessoas detentoras destes cargos atuem sobre o sistema em atividades como “Elaborar ementa”, “Registrar cuidados especiais”, ou, ainda, “Calendarizar limpezas aos sanitários”, entre outras. Efetivamente, parece ter havido, por parte da equipe que elaborou a documentação original, algum equívoco entre “ter direito de acesso à determinada informação” e “executar todas as atividades relativas àquela mesma informação”. De todo modo, a Matriz de Responsabilidades elaborada a partir da nova WBS (Anexo 4), apresenta uma distribuição mais alinhada a um provável cenário de uso real do referido sistema.

Por fim, diante de todo o exposto neste capítulo e a título de encerramento das considerações sobre este Estudo de Caso, observa-se que os resultados obtidos mostraram-se positivos, tendo em vista que ficou evidenciada a aplicabilidade de técnicas de Gestão de Projetos para o levantamento e caracterização de requisitos em sistemas orientados a objetos.

6. Conclusões e trabalhos futuros

Do que foi apresentado ao longo desta dissertação, percebe-se a importância do levantamento de requisitos em desenvolvimento de sistemas e de como tal fase constitui-se num fator impactante no sucesso de um projeto desta natureza. Assim, ao se introduzir o presente estudo, procurou-se ressaltar a necessidade de os requisitos serem elaborados da forma mais completa e coesa possível, ainda nas fases iniciais, obedecendo regras e critérios definidos, procurando evitar a amplificação de erros em fases posteriores. Nesse sentido, estatísticas relevantes foram apresentadas, corroborando a percepção de que os sistemas informáticos atuais ainda carecem de maior robustez no que diz respeito ao levantamento e definição de requisitos.

Por outro lado, é bastante conhecida a maturidade da área de Gestão de Projetos, particularmente no tocante à questão da definição de âmbito. Seu sucesso, muito relacionado à aplicação extensiva do Guia PMBoK, editado pelo *Project Management Institute* – PMI, ampara-se na utilização sistemática de técnicas que facilitam a obtenção e gestão das atividades necessárias à execução de um projeto qualquer.

Estes dois contextos, associados ao fato de que o desenvolvimento de um sistema também é um projeto, conduziu esta autora à indagação de partida deste trabalho, ou seja, se seria possível o uso de técnicas de definição de âmbito, da área de Gestão de Projetos, para uma melhor definição de requisitos, em sistemas orientados a objetos.

Assim, a fim de se obter uma resposta a esta questão, foram estabelecidos três objetivos para o presente estudo, a saber: a) analisar as similaridades conceituais entre as áreas de Gestão de Projetos e de Análise Orientada a Objetos, a partir de seus respectivos documentos de definição de âmbito/requisitos; b) propor um processo de transposição de técnicas de definição de âmbito, em Gestão de Projetos, devidamente adaptados ao contexto de análise orientada a objetos, o qual permita uma elaboração mais abrangente de artefatos em levantamento de requisitos; e c) aplicar o processo proposto em um sistema orientado a objetos existente, afim de se experimentar a sua viabilidade prática.

Desta forma, ao se analisar a dimensão de Análise de Sistemas, foi discutida a sua evolução, desde as Análises Estruturada e Essencial, até a Análise Orientada a Objetos, sendo esta, por sua vez, considerada a mais utilizada, atualmente, no desenvolvimento de sistemas. Por esta razão, foram apresentados os principais artefatos recomendados pela metodologia *Rational Unified Process* - RUP, a qual está amplamente apoiada na linguagem *Unified Modeling Language* – UML – para sistemas orientados a objetos. Neste caso, foi notado que há um grande número de artefatos a serem produzidos

durante o ciclo de vida, o que pode estar dificultando a percepção das funcionalidades do *software* a ser desenvolvido, bem como a comunicação entre desenvolvedores e clientes.

Por outro lado, ao se analisar a área de Gestão de Projetos, observou-se a maturidade da mesma, particularmente no tocante à questão da definição de âmbito. Nesse contexto, diversos documentos são utilizados para este fim, de forma que o presente estudo destacou alguns produzidos nas fases de Iniciação e Planejamento, os quais têm o objetivo de levantar e delimitar as atividades a serem executadas a fim de se garantir o sucesso de um projeto. Deste modo, estes foram analisados e exemplificados com o intuito de serem experimentados na dimensão de desenvolvimento de sistemas orientados a objetos.

A partir destas análises e de um entendimento mais apurado, bem como no seguimento dos objetivos propostos, foi discutida e apresentada uma estratégia de conexão conceitual entre as dimensões supracitadas, no que diz respeito a requisitos/âmbito. A análise efetuada para tanto fundamentou-se em duas questões comuns a ambas as áreas, a saber: *o quê* e *quem*. Observou-se, desta feita, que as duas áreas possuem documentos e artefatos bastante similares que tratam daqueles aspectos, o que apontou para uma possível conexão conceitual entre ambas. Neste sentido, observou-se que a utilização de técnicas como Declaração de Âmbito, *Work Breakdown Structure* (WBS) e Matriz de Responsabilidades, podem fornecer inúmeros *inputs* relevantes para diversos artefatos típicos num desenvolvimento de sistemas orientado a objetos, nomeadamente os Diagramas de Casos de Uso e Diagrama de Classes.

Por outro lado, apesar de se verificar tal possibilidade, percebeu-se a necessidade de adequações dos documentos de Gestão de Projetos que poderiam ser emprestados à área de Análise Orientada a Objetos, conforme discutido na introdução do capítulo 4, principalmente por conta de seus diferentes enfoques. De todo o modo, feitos os ajustes e eliminações necessários, foi possível elaborar e propor um processo de uso de tais documentos, o qual está constituído em cinco etapas, as quais viabilizam uma ligação instrumental entre aquelas áreas.

No entanto, a simples possibilidade de tal conexão conceitual não era suficiente para se experimentar a viabilidade prática desta abordagem, razão pela qual realizou-se um Estudo de Caso, conforme o último objetivo definido para este trabalho. Deste modo, a partir da documentação relativa a um sistema existente, a qual foi elaborada utilizando-se a abordagem orientada a objetos, aplicou-se o processo proposto no Capítulo 4. Desta forma, foi possível efetuarem-se comparações entre os novos artefatos produzidos e os anteriormente documentados.

Diante dos resultados obtidos, observou-se que, para além de oferecer uma técnica viável e prática de delineamento de requisitos, a abordagem sugerida apresenta outras vantagens dignas de serem

mencionadas. Destaque-se, por exemplo, o uso da Declaração de Âmbito, conforme estruturada no tópico 4.3, a qual, além de fornecer informações sobre objetivos e âmbito do sistema, suas fronteiras ambientais, restrições e pressupostos, também traz uma lista dos principais atores (ou papéis) que farão uso do sistema a ser desenvolvido. Esses atores, por sua vez, ao tornarem-se atores nos Casos de Uso elaborados em etapa posterior, acabaram por facilitar sobremaneira a elaboração daquele artefato, além de propiciar a eliminação de atores passivos, ou seja, que não operavam sobre o sistema.

O uso da WBS, por sua vez, ao ser elaborada com o enfoque ajustado ao que foi discutido no tópico 4.1, assemelha-se muito a um organograma, o qual é de fácil compreensão e amplamente utilizado em várias organizações. Nesse sentido, permite ao cliente, bem como à equipe desenvolvedora, a visão do sistema como um todo, o que, por sua vez, facilita a comunicação entre os dois lados. Para além disso, com o uso da técnica de WBS, todos os envolvidos têm condições de observar, num único diagrama, todas as partes constituintes do sistema, seja ao nível macro, onde se notam os temas/assuntos que o sistema trata, seja ao nível micro, onde vêem-se as atividades funcionais específicas como inclusões, suspensões, exclusões e edições, entre outras, de acordo com o tipo de sistema. De fato, em consequência da análise e discussão com o cliente sobre as atividades que deverão constar na WBS, em conjunto com os atores listados na Declaração de Âmbito, torna-se possível à equipe desenvolvedora elaborar casos de uso coerentes e funcionalmente alinhados.

Outra consequência que desperta interesse ao se utilizar a WBS é que esta pode facilitar a validação, por parte do cliente, dos requisitos que serão implementados pelo sistema que se está a desenvolver. Nesse mesmo sentido, a WBS permite ao cliente a correta percepção dos requisitos, sua localização dentro do sistema e sua integração com os demais módulos funcionais. Mais ainda, a WBS permite que o cliente critique, objetivamente, os requisitos a serem atendidos e, por conseguinte, eventuais inclusões de novos requisitos, ou, ainda, edições e mesmo exclusões de requisitos previamente acordados.

Do mesmo modo, tendo em vista que a WBS deverá ser elaborada em conjunto com o cliente, pressupõe-se que a definição dos temas/assuntos do sistema, os quais constituir-se-ão nos seus módulos funcionais de mais alto nível, terão um caráter mais intuitivo e de fácil compreensão por parte dos utilizadores finais. Tal característica mostra-se interessante, pois, nas metodologias usuais de desenvolvimento de sistemas orientados a objetos, tal tarefa fica normalmente a cargo da equipe desenvolvedora, o que pode implicar numa decomposição do sistema em módulos funcionais que fazem

pouco sentido ao cliente e/ou seus utilizadores. Diante deste contexto, observa-se, então, que a WBS pode facilitar bastante a interação entre equipe e cliente acerca dos requisitos a serem atendidos.

O uso da Matriz de Responsabilidades, por sua vez, ao ser elaborada a partir da WBS feita em etapa anterior, trouxe outro conjunto de características que podem revelar-se bastante úteis. Primeiramente, há que se observar que a aplicação da abordagem sugerida em 4.3, ao Estudo de Caso utilizado, deixou clara a eficiência do processo estruturado e coordenado de descoberta das classes funcionais de um sistema, bem como de nomeação e distribuição coerente de seus métodos, além de um critério prático para a definição das associações entre aquelas mesmas classes. Estas características, em conjunto, facilitaram bastante a elaboração do Diagrama de Classes inicial do sistema no Estudo de Caso realizado.

Em segundo lugar, ainda relativamente à Matriz de Responsabilidades, para além de esta propiciar mais uma uma visão global dos requisitos do sistema, ela não só fornece informações cruciais sobre os principais papéis (ou atores) envolvidos no uso rotineiro do sistema em questão, como também a atuação de cada um deles sobre o mesmo. Em realidade, tais características têm a propriedade de fornecer aos envolvidos (membros da equipe desenvolvedora e cliente) uma percepção imediata de todos os perfis de acesso ao sistema, com implicações óbvias nas discussões sobre a segurança da informação a ser implementada sobre o mesmo. Nesse sentido, o Estudo de Caso apresentado foi particularmente adequado, posto que, de fato, informações acerca de pacientes hospitalares têm caráter confidencial, com elevado grau de confidencialidade.

No que diz respeito a possíveis estudos futuros, a abordagem proposta apresenta uma característica que lhe é peculiar, mas que, no entanto, pode se tornar uma ressalva importante, pelo que sugere-se investigação específica. Ocorre que as referidas técnicas, por terem sido emprestadas da área de Gestão de Projetos, partem do princípio de que todos os atores são humanos, o que implica afirmar que estes atuarão sobre o sistema com graus de liberdade e de tempo próprios. Deste modo, talvez tal abordagem não se mostre prática, ou mesmo viável, para sistemas de tempo real ou que tenham forte interação com outros sistemas, ou mesmo equipamentos. Uma provável razão para tanto, ainda a ser verificada, é que o contexto de utilização de suas funcionalidades pode ser absolutamente caótico, com uma infinidade de entradas sendo analisadas simultaneamente e respondidas de acordo, como, por exemplo, nos chamados “sistemas embarcados”, como aqueles existentes em aeronaves, veículos não tripulados ou, ainda, sistemas de controle de operações bancárias, entre outros.

Por fim, diante de todos os resultados e percepções apresentados, o presente trabalho mostrou ser viável a utilização de técnicas de gestão de projetos para o levantamento e caracterização de requisitos para o desenvolvimento de sistemas orientados a objetos. Entende-se que, se estas forem aplicadas com o foco exclusivo no sistema e seus requisitos funcionais, as mesmas podem produzir um conjunto de artefatos coerentes e alinhados entre si, que auxiliarão na produção de sistemas com maior grau de qualidade final.

Referências

- Aken, J. E. (2005). Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management. *British Journal of Management, Vol.16, n° 1*, pp. 19-36.
- APM.Group. (2012, maio 10). *PRINCE2 - PRojects IN Controlled Environments*. Retrieved from <http://www.prince-officialsite.com/home/home.aspx>
- Arantes, E., Anselmo, J., Senise, L., & Sabinelli, P. (2008). *Gerenciamento de projetos*. http://www.promon.com.br/portugues/noticias/download/PBTR%20GE_para%20web.pdf. Brasil: Promom, Business & Tecnology Review.
- Aurum, A., & Wohlin, C. (2005). *Engineering and Managing Software Requirements*. USA: Springer.
- Avgerou, C., & McGrath, K. (2007). Power, rationality, and the art of living through socio-technical change. *MIS quarterly, Vol. 31, n° 2*, pp. 293-315.
- Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2008). *Thesis Projects. A Guide for Students in Computer Science and Information Systems. 2ª Ed.* England: Springer.
- Boehm, B. (1996, July). Anchoring the Software Process. *IEEE Software. Vol. 13, n.4*, pp. 73-82.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide*. USA: Addison-Wesley Longman Inc.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide. Second Edition*. USA: Addison Wesley.
- Cerpa, N., & Verner, J. M. (2009). Why did your project fail? *Science Education. Communications of the ACM Vol. 52 n°12*, pp. 130-134.
- Chiu, C. M. (2005). Applying means-end chain theory to eliciting system requirements and understanding users perceptual orientations. *Information and Management, Vol. 42, Elsevier*, pp. 455-468.
- CIDA. (2012, maio 10). *Canadian International Development Agency*. Retrieved from <http://www.acdi-cida.gc.ca/rbm>

- Dalmazo, L. (2008). *Um em cada três projetos de TI não atinge objetivos, diz estudo*. Retrieved from ComputerWorld: <http://computerworld.uol.com.br/gestao/2008/01/10/idgnoticia.2008-01-10.9549661774/>
- DeMarco, T. (2001). Structured Analysis: Beginnings of a New Discipline. In M. Broy, & E. D. (Eds.), *Software Pioneers - Contributions to Software Engineering* (pp. 520-528). Germany: Springer.
- Dicionário Universal. (2009). *Dicionário Fundamental da Língua Portuguesa*. Portugal: Texto Editores, Lda.
- Dinsmore, P. C., & Cavelleri, A. (2008). *Gerenciamento de Projetos. 2ª Ed.* Brasil: Qualitymark.
- Dobing, B., & Parsons, J. (2008). Dimensions of UML Diagram Use: A Survey of Practitioners, Vol.19(1). pp. 1-18.
- Eveleens, J. L., & Verhoef, C. (2010). The rise and fall of the Chaos report figures. *Focus Project Management. Published by the IEEE Computer Society*, pp. 30-36.
- Finkelstein, L., Huang, J., Finkelstein, A. C., & Nuseibeh, B. (1992, Apr-Jun). Using Software Specification Methods for Measurement Instrument Systems. *Measurement Journal, Vol 10, No 2*, pp. 79-86.
- Gartner, I. (2011, abril 14-15). *Gartner Says Worldwide Application Infrastructure and Middleware Market Revenue Increased 7.3 Percent in 2010*. Retrieved from Gartner Newsroom: <http://www.gartner.com/it/page.jsp?id=1632714>
- Guedes, G. T. (2005). *UML 2. Guia de consulta rápida*. Brasil: Novatec Ltda.
- IBM. (2012). *IBM Rational Unified Process*. Retrieved from Proven best practices for software and systems delivery and implementation and for effective project management: <http://www-01.ibm.com/software/awdtools/rup/>
- IPMA. (2012, maio 10). *IPMA - International Project Management Association*. Retrieved from <http://ipma.ch/certification/competence/ipma-competence-baseline/>
- ISRD.Group. (2007). *Structured System Analysis and Design*. India: Tata Mc_Graw Hill.
- IT.Web. (2008). *Falta de recursos em TI é problema mundial*. Retrieved from <http://itweb.com.br/30869/falta-de-recursos-em-ti-e-problema-mundial/>

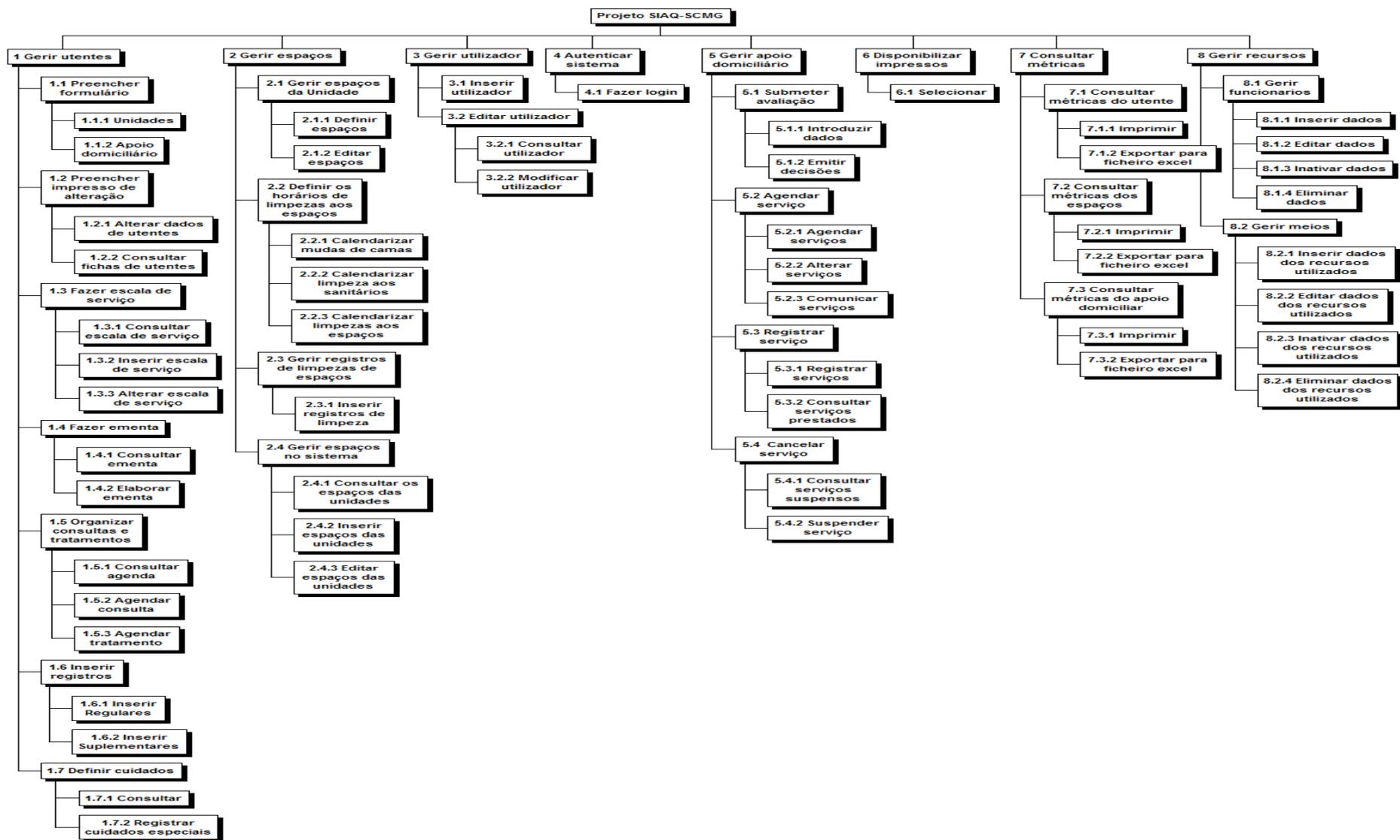
- Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *Unified software development process, the complete guide to the unified process from the original designers*. USA: Addison Wesley Longman Inc.
- Jones, C. (1994). *Assessment and control of software risks*. USA: Yourdon Press Upper Saddle River.
- Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. England: John Wiley and Sons Ltd.
- Kulak, D., & Guiney, E. (2001). *Use Cases: Requirements in Context*. New York: Addison Wesley.
- Larman, C. (2008). *Utilizando UML e Padrões. Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo*. Brasil: Bookman.
- Lopes, F. C., Morais, M. P., & Carvalho, A. J. (2009). *Desenvolvimento de Sistemas de Informação. 2ª Edição*. Portugal: FCA - Editora de Informática, Lda.
- Lucas, J., Gomes, M., Lemos, H., Neiva, R., Fernandes, F., Teixeira, M., . . . Lopes, P. (2011). *Sistema de Apoio a Qualidade - Santa Casa de Misericórdia de Guimarães. Disponibilizado pelo Prof. Dr. Ricardo J. Machado*. Portugal.
- Manson, N. (2006). Is operations research really research? *Operations Research Society of South África*, Vol 22(2). pp. 155–180.
- Martinsons, M. G., & Chong, P. K. (1999). The Influence of Human Factors and Specialist Involvement on Information Systems Success. *Humam Relations, Vol 52, n° 1, Springer*, pp. 123-152.
- Mecenas, I., & Oliveira, V. (2005). *Qualidade em Software – Uma metodologia para homologação de sistemas*. Brasil: Alta Books.
- Miguel, A. (2009). *Gestão Moderna de Projectos. Melhores Técnicas e Práticas. 5ª Ed.* Portugal: FCA - Editora Informática.
- Miguel, A. (2010). *Gestão de Projectos de Software. 4ª Ed.* Portugal: FCA, Editora de Informática, Lda.
- Nunes, M., & O'Neill, H. (2008). *Fundamental de UML. 6ª Edição*. Portugal: FCA - Editora de Informática, Lda.
- Parker, A. (2000). Commons Committee Calls for Action on IT Fiascos. *Financial Times, London*, 2.

- PM4DV. (2007). *Fundamentals of Project Management. A modern methodology to manage development projects for international assistance and humanitarian relief organizations*. USA: Project Management For Development Organizations.
- PMBok. (2008). *A Guide to the Project Management Body of Knowledge. 4ª Ed*. EUA: Project Management Institute, Inc.
- PMI. (2012, maio 12). *Project Management Institute*. Retrieved from <http://www.pmi.org/About-Us.aspx>
- Pressman, R. S. (1995). *Software Engineering: Practitioner's Approach. 3ª Ed*. England: McGraw-Hill.
- Pressman, R. S. (1997). *Software Engineering - A practitioner's approach. Fourth Edition*. USA: McGraw-Hill.
- Priberam. (2012, Julho 18). *Dicionário Priberam da Língua Portuguesa*. Retrieved from <http://www.priberam.pt/dlpo/default.aspx?pal=processo>
- Project Management Institute. (2006). *Practice Standard for Work Breakdown Structures - Second Edition*. USA: Project Management Institute, Inc.
- Rational. (2006). *RUP for small projects*. Retrieved from WThreeX: <http://www.wthree.com/rup/smallprojects/index.htm>
- Robb, D. (2009). *Why Software Development Projects Fail*. Retrieved from ITBusinessEdge. CIO Update. Community for Information Technology Leaders: <http://www.cioupdate.com/insights/article.php/3801091/Why-Software-Development-Projects-Fail-Part-I.htm>
- Satzinger, J. W. (1993). Introduction to Essential Systems Analysis. *Proceedings of the Seventh Annual Conference of the International Academy for Information Management*, pp. 1-13.
- SCMG. (2012, Abril 26). *Santa Casa da Misericórdia de Guimarães*. Retrieved from <http://www.scmguimaraes.com/index.php>
- Silva, A. M., & Videira, C. A. (2001). *UML - Metodologias e Ferramentas CASE*. Portugal: Centro Atlântico, Lda.
- Sommerville, I. (2007). *Engenharia de Software. 8ª Ed*. Brasil: Addison Wesley.

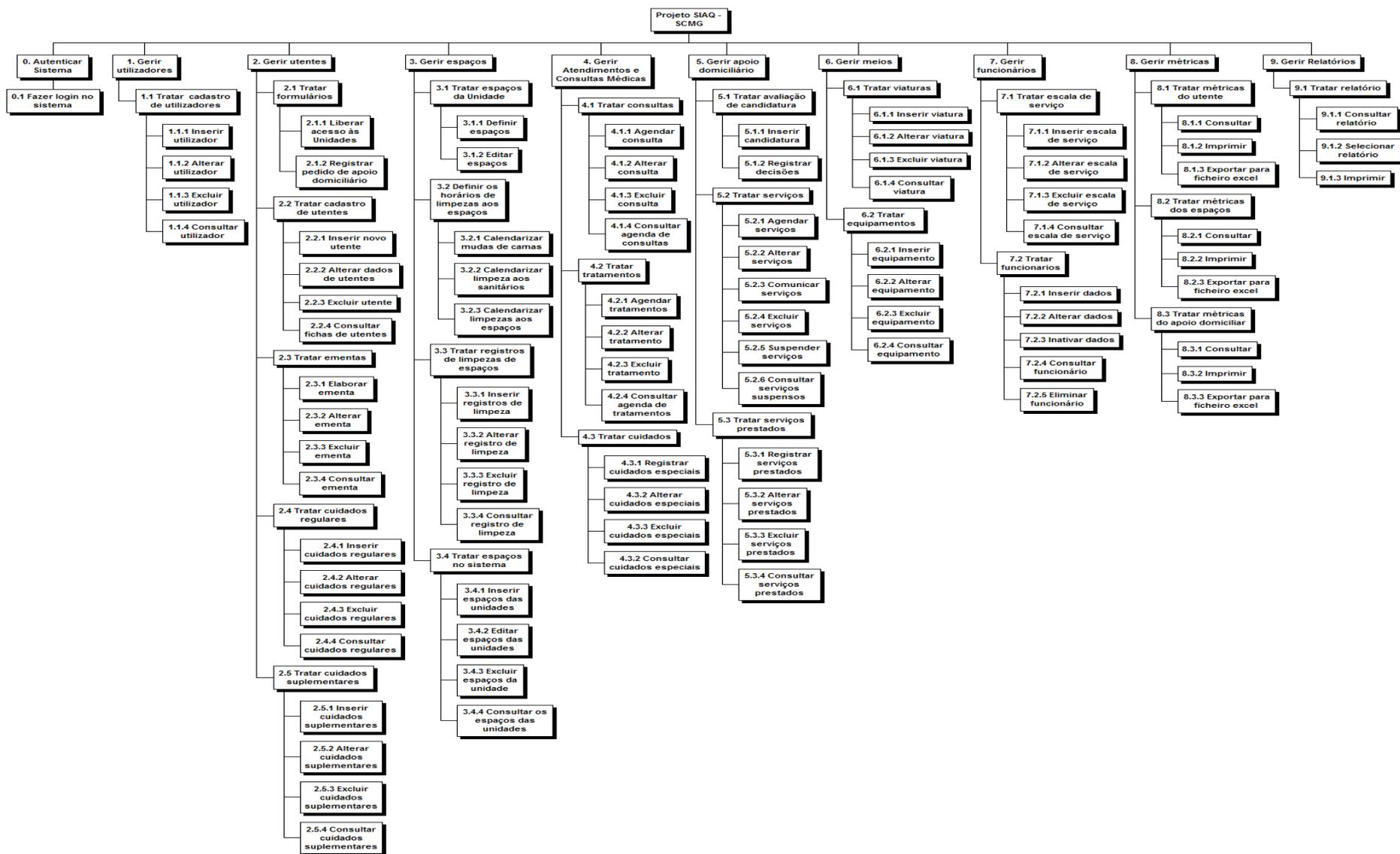
- Stair, R., & Reynolds, G. (2008). *Principles of Information Systems. 8ª Ed.* EUA: McGraw Hill.
- Standing, C., Guilfoyle, A., Lin, C., & Love, P. E. (2006). The attribution of success and failure in IT projects. *Industrial Management & Data Systems. Emerald Group Publishing Limited. Vol. 106, n° 8*, pp. 1148 - 1165.
- Taylor, J. C. (2008). *Project Scheduling and Cost Control: Planning, Monitoring and Controlling the Baseline.* USA: J. Ross Publishing, Inc.
- Tonsig, S. L. (2008). *Engenharia de Software – Análise e Projeto de Sistemas. 2 ed.* Brasil: Editora Ciência Moderna Ltda.
- Wastell, D., Sauer, J., & Schmeink, C. (2009). Time for a "design turn" in IS innovation research? A practice report from the home front. *Information Technology & People, 22(4)*, pp. 335-349.
- ZOPP. (2012, maio 10). *International Development Cooperation / Consultants without Borders.* Retrieved from ZOPP Ziel-Orientierte Projekt Planung: <http://www.weitzenegger.de/en/zopp.html>

ANEXOS

Anexo 1 - WBS elaborada a partir dos requisitos funcionais da documentação original



Anexo 2 - WBS obtida após aplicação da técnica proposta



Anexo 3 - Matriz de responsabilidades elaborada a partir dos requisitos funcionais da documentação original

| SIAQ - SCMG | | | | | | | | | | |
|---|--|--|--------------|----------------|----------------|----------|--------|------------|---------------|------------|
| Tema/ assunto | Atividades | Responsabilidades | Papéis | | | | | | | |
| | | | Dir. Unidade | Dir. Qualidade | Administrativo | Provedor | Médico | Enfermeira | Aux. de saúde | Impressora |
| 1. Gerir utentes | 1.1. Preencher formulário | 1.1.1. Acesso às Unidades 1.1.2. Apoio domiciliário | | | | | | | | |
| | 1.2. Preencher impresso de alteração | 1.2.1. Alterar dados | | | | | | | | |
| | | 1.2.2. Consultar fichas de utentes | | | | | | | | |
| | 1.3. Fazer escala de serviço | 1.3.1. Consultar escala de serviço | | | | | | | | |
| | | 1.3.2. Inserir escala de serviço | | | | | | | | |
| | | 1.3.3. Alterar escala de serviço | | | | | | | | |
| | 1.4. Fazer ementas | 1.4.1. Consultar ementa | | | | | | | | |
| | | 1.4.2. Elaborar ementa | | | | | | | | |
| 1.5. Organizar consultas e tratamentos | 1.5.1. Consultar agenda | | | | | | | | | |
| | 1.5.2. Agendar consulta | | | | | | | | | |
| | 1.5.3. Agendar tratamento | | | | | | | | | |
| 1.6. Inserir registos | 1.6.1. Inserir regulares | | | | | | | | | |
| | 1.6.2. Inserir suplementares | | | | | | | | | |
| 1.7. Definir cuidados | 1.7.1. Consultar | | | | | | | | | |
| | 1.7.2. Registrar cuidados especiais | | | | | | | | | |
| 2. Gerir espaços | 2.1. Gerir espaços da unidade | 2.1.1. Definir espaços | | | | | | | | |
| | | 2.1.2. Editar espaços | | | | | | | | |
| | 2.2. Definir horários de limpeza aos espaços | 2.2.1. Calendarizar mudas de camas | | | | | | | | |
| | | 2.2.2. Calendarizar limpezas aos sanitários | | | | | | | | |
| | 2.2.3. Calendarizar limpezas aos espaços | | | | | | | | | |
| 2.3. Gerir registos de limpeza aos espaços | 2.3.1. Inserir registos de limpeza | | | | | | | | | |
| | 2.3.2. Consultar os espaços da unidade | | | | | | | | | |
| 2.4. Gerir espaços no sistema | 2.4.1. Inserir espaços da unidade | | | | | | | | | |
| | 2.4.2. Editar espaços da unidade | | | | | | | | | |
| 3. Gerir utilizadores | 1.1. Inserir utilizador | — | | | | | | | | |
| | 1.2. Editar utilizador | 1.2.1. Consultar utilizador | | | | | | | | |
| 1.2.2. Modificar utilizador | | | | | | | | | | |
| 4. Autenticar sistema | 0.1. Fazer login | — | | | | | | | | |
| 5. Gerir apoio domiciliário | 5.1. Submeter avaliação | 5.1.1. Introduzir dados | | | | | | | | |
| | | 5.1.2. Emitir decisões | | | | | | | | |
| | 5.2. Agendar serviços | 5.2.1. Agendar serviços | | | | | | | | |
| | | 5.2.2. Alterar serviços | | | | | | | | |
| | 5.2.3. Comunicar serviços | | | | | | | | | |
| 5.3. Registrar serviços | 5.3.1. Registrar serviços | | | | | | | | | |
| | 5.3.2. Consultar serviços prestados | | | | | | | | | |
| 5.4. Cancelar serviço | 5.2.4. Suspender serviços | | | | | | | | | |
| | 5.2.5. Consultar serviços suspensos | | | | | | | | | |
| 6. Disponibilizar impressos | 6.1 Selecionar | — | | | | | | | | |
| 7. Consultar métricas | 7.1. Consultar métricas do utente | 8.1.2. Imprimir | | | | | | | | |
| | | 8.1.3. Exportar para ficheiro Excel | | | | | | | | |
| | 7.2. Consultar métricas do espaço | 8.2.2. Imprimir | | | | | | | | |
| | | 8.2.3. Exportar para ficheiro Excel | | | | | | | | |
| 8.3. Consultar métricas de apoio domiciliário | 8.3.2. Imprimir | | | | | | | | | |
| | 8.3.3. Exportar para ficheiro Excel | | | | | | | | | |
| 8. Gerir recursos | 8.1. Gerir funcionários | 8.1.1. Inserir dados | | | | | | | | |
| | | 8.1.2. Editar dados | | | | | | | | |
| | | 8.1.3. Inativar dados | | | | | | | | |
| | | 8.1.4. Eliminar funcionário | | | | | | | | |
| | 8.2 Gerir meio | 8.1.1. Inserir dados dos recursos utilizados | | | | | | | | |
| | | 8.1.1. Editar dados dos recursos utilizados | | | | | | | | |
| | | 8.1.3. Inativar dados dos recursos utilizados | | | | | | | | |
| | | 8.1.4. Eliminar dados dos recursos utilizados | | | | | | | | |

Anexo 4 - Matriz de responsabilidades decorrente da aplicação da proposta

| SIAQ - SCMG | | | | | | | | |
|---|--|---|--------------|----------------|----------------|----------|--------|------------|
| Tema/ assunto | Atividades | Responsabilidades | Papéis | | | | | |
| | | | Dir. Unidade | Dir. Qualidade | Administrativo | Provedor | Médico | Enfermeira |
| 0. Autenticar sistema | 0.1. Fazer login | 0.1.1. Inserir dados pessoais | | | | | | |
| 1. Gerir utilizadores | 1.1. Tratar cadastro de utilizadores | 1.1.1. Inserir novo utilizador | | | | | | |
| | | 1.1.2. Alterar utilizador | | | | | | |
| | | 1.1.3. Excluir utilizador | | | | | | |
| | | 1.1.4. Consultar utilizador | | | | | | |
| 2. Gerir utentes | 2.1. Tratar formulários | 2.1.1. Liberar acesso às unidades | | | | | | |
| | | 2.1.2. Registrar pedido de apoio domiciliário | | | | | | |
| | 2.2. Tratar cadastro de utentes | 2.2.1. Inserir novo utente | | | | | | |
| | | 2.2.2. Alterar dados de utentes | | | | | | |
| | | 2.2.3. Excluir utente | | | | | | |
| | | 2.2.4. Consultar fichas de utentes | | | | | | |
| | 2.3. Tratar ementas | 2.3.1. Elaborar ementa | | | | | | |
| | | 2.3.2. Alterar ementa | | | | | | |
| | | 2.3.3. Excluir ementa | | | | | | |
| | | 2.3.4. Consultar ementa | | | | | | |
| | 2.4. Tratar cuidados regulares | 2.4.1. Inserir cuidados regulares | | | | | | |
| | | 2.4.2. Alterar cuidados regulares | | | | | | |
| | | 2.4.3. Excluir cuidados regulares | | | | | | |
| | | 2.4.4. Consultar cuidados regulares | | | | | | |
| | 2.5. Tratar cuidados suplementares | 2.5.1. Inserir cuidados suplementares | | | | | | |
| | | 2.5.2. Alterar cuidados suplementares | | | | | | |
| 2.5.3. Excluir cuidados suplementares | | | | | | | | |
| 2.5.4. Consultar cuidados suplementares | | | | | | | | |
| 3. Gerir espaços | 3.1. Tratar espaços da unidade | 3.1.1. Definir espaços | | | | | | |
| | | 3.1.2. Editar espaços | | | | | | |
| | 3.2. Definir horários de limpeza aos espaços | 3.2.1. Calendarizar mudas de camas | | | | | | |
| | | 3.2.2. Calendarizar limpezas aos sanitários | | | | | | |
| | | 3.2.3. Calendarizar limpezas aos espaços | | | | | | |
| | 3.3. Tratar registos de limpeza aos espaços | 3.3.1. Inserir registo de limpeza | | | | | | |
| | | 3.3.2. Alterar registo de limpeza | | | | | | |
| | | 3.3.3. Excluir registo de limpeza | | | | | | |
| | | 3.3.4. Consultar registo de limpeza | | | | | | |
| | 3.4. Tratar espaços no sistema | 3.4.1. Inserir espaços da unidade | | | | | | |
| | | 3.4.2. Editar espaços da unidade | | | | | | |
| | | 3.4.3. Excluir espaços da unidade | | | | | | |
| 3.4.4. Consultar espaços da unidade | | | | | | | | |
| 4. Gerir Atendimentos e consultas médicas | 4.1. Tratar consultas | 4.1.1. Agendar consulta | | | | | | |
| | | 4.1.2. Alterar consulta | | | | | | |
| | | 4.1.3. Excluir consulta | | | | | | |
| | | 4.1.4. Consultar agenda de consultas | | | | | | |
| | 4.2. Tratar tratamentos | 4.2.1. Agendar tratamento | | | | | | |
| | | 4.2.2. Alterar tratamento | | | | | | |
| | | 4.2.3. Excluir tratamento | | | | | | |
| | | 4.2.4. Consultar agenda de tratamentos | | | | | | |
| | 4.3. Tratar cuidados | 4.3.1. Registrar cuidados especiais | | | | | | |
| | | 4.3.2. Alterar cuidados especiais | | | | | | |
| | | 4.3.3. Excluir cuidados especiais | | | | | | |
| | | 4.3.4. Consultar cuidados especiais | | | | | | |

(Anexo 4 - continuação)

| | | | | | | | | | |
|-------------------------------------|--|-------------------------------------|--|--|--|--|--|--|--|
| 5. Gerir apoio domiciliário | 5.1. Tratar avaliação de candidatura | 5.1.1. Inserir candidatura | | | | | | | |
| | | 5.1.2. Registrar decisão | | | | | | | |
| | 5.2. Tratar serviços | 5.2.1. Agendar serviços | | | | | | | |
| | | 5.2.2. Alterar serviços | | | | | | | |
| | | 5.2.3. Comunicar serviços | | | | | | | |
| | | 5.2.4. Excluir serviços | | | | | | | |
| | | 5.2.5. Suspender serviços | | | | | | | |
| | | 5.2.6. Consultar serviços suspensos | | | | | | | |
| | 5.3. Tratar serviços prestados | 5.3.1. Registrar serviços prestados | | | | | | | |
| | | 5.3.2. Alterar serviços prestados | | | | | | | |
| 5.3.3. Excluir serviços prestados | | | | | | | | | |
| 5.3.4. Consultar serviços prestados | | | | | | | | | |
| 6. Gerir meios | 6.1. Tratar viaturas | 6.1.1. Inserir viatura | | | | | | | |
| | | 6.1.2. Alterar viatura | | | | | | | |
| | | 6.1.3. Excluir viatura | | | | | | | |
| | | 6.1.4. Consultar viatura | | | | | | | |
| | 6.2. Tratar equipamentos | 6.2.1. Inserir equipamento | | | | | | | |
| | | 6.2.2. Alterar equipamento | | | | | | | |
| | | 6.2.3. Excluir equipamento | | | | | | | |
| | | 6.2.4. Consultar equipamento | | | | | | | |
| 7. Gerir funcionário | 7.1. Tratar escala de serviço | 7.1.1. Inserir escala de serviço | | | | | | | |
| | | 7.1.2. Alterar escala de serviço | | | | | | | |
| | | 7.1.3. Excluir escala de serviço | | | | | | | |
| | | 7.1.4. Consultar escala de serviço | | | | | | | |
| | 7.2. Tratar funcionários | 7.2.1. Inserir dados | | | | | | | |
| | | 7.2.2. Alterar dados | | | | | | | |
| | | 7.2.3. Inativar dados | | | | | | | |
| | | 7.2.4. Consultar funcionário | | | | | | | |
| | | 7.2.5. Eliminar funcionário | | | | | | | |
| 8. Gerir métricas | 8.1. Tratar métricas do utente | 8.1.1. Consultar | | | | | | | |
| | | 8.1.2. Imprimir | | | | | | | |
| | | 8.1.3. Exportar para ficheiro Excel | | | | | | | |
| | 8.2. Tratar métricas do espaço | 8.2.1. Consultar | | | | | | | |
| | | 8.2.2. Imprimir | | | | | | | |
| | | 8.2.3. Exportar para ficheiro Excel | | | | | | | |
| | 8.3. Tratar métricas de apoio domiciliário | 8.3.1. Consultar | | | | | | | |
| | | 8.3.2. Imprimir | | | | | | | |
| | | 8.3.3. Exportar para ficheiro Excel | | | | | | | |
| 9. Gerir relatórios | 9.1. Tratar relatório | 9.1.1. consultar relatório | | | | | | | |
| | | 9.1.2. Selecionar relatório | | | | | | | |
| | | 9.1.3. Imprimir | | | | | | | |