

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

LUIZ CARLOS MOSER

**FERRAMENTA PARA IMPLANTAR A METODOLOGIA MULTICRITÉRIO DE
APOIO À DECISÃO – CONSTRUTIVISTA**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

LUIZ CARLOS MOSER

**FERRAMENTA PARA IMPLANTAR A METODOLOGIA MULTICRITÉRIO DE
APOIO À DECISÃO – CONSTRUTIVISTA**

Trabalho de Conclusão de Curso de graduação, apresentado à banca do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

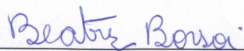
Orientadora: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2015**

ATA Nº: 004

DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO DO ALUNO LUIZ CARLOS MOSER.

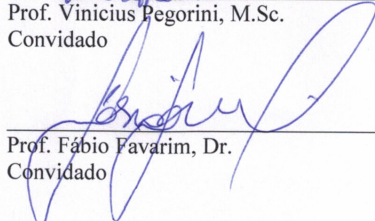
Às 17:45 do dia 24 de novembro de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (orientadora), Vinicius Pegorini (convidado) e Fábio Favarim (convidado), para avaliar o Trabalho de Conclusão de Curso do aluno Luiz Carlos Moser, matrícula 1494112, sob o título **Ferramenta para implantar a metodologia multicritério de apoio à decisão – construtivista**; como requisito final para a conclusão da disciplina Trabalho de Conclusão de Curso 2 do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 18:40 hrs foi encerrada a sessão.



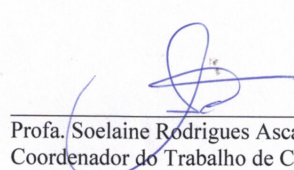
Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



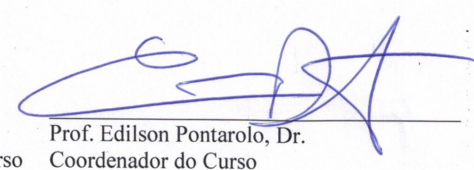
Prof. Vinicius Pegorini, M.Sc.
Convidado



Prof. Fábio Favarim, Dr.
Convidado



Profa. Soelaine Rodrigues Ascari, M.Sc.
Coordenador do Trabalho de Conclusão de Curso



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

Não existem pessoas especiais, tudo é resultado do esforço e da dedicação.

DEDICATÓRIA

Dedico este trabalho de TCC a minha esposa Rejane Canton Moser que de forma especial e carinhosa me deu força e coragem, me apoiando nos momentos de dificuldades e a minha filha Phamela Cristina Moser pelo apoio e confiança.

AGRADECIMENTOS

A minha esposa Rejane Canton Moser pela paciência e compreensão na correria de cada semestre dedicado ao estudo.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

À minha orientadora profa. Beatriz T. Borsoi, pela dedicação e suporte no tempo que lhe coube, pelas suas correções e incentivos.

Ao prof. Sandro Bortoluzzi pelo tempo disponibilizado e pelo compartilhamento do conhecimento que foi fundamental para a construção dos casos de uso deste trabalho.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

MOSER, Luiz Carlos. Ferramenta para implantar a metodologia multicritério de apoio à decisão – construtivista. 2015. 57f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

A tomada de decisão em relação à aplicação de recursos pode ser dependente de muitos critérios e variáveis. Esse processo se torna mais crítico à medida que os recursos são mais escassos. Em Instituições públicas como Universidades, a aplicação de recursos sempre é uma tarefa de decisão difícil. A multiplicidade de interesses e necessidades envolvidos na aplicação de recursos torna esse processo trabalhoso e difícil. A metodologia Multicritério de Apoio à Decisão – Construtivista (MCDA-C) visa fornecer um processo de apoio à decisão baseado em interesses e critérios do decisor. A complexidade da própria metodologia e a multiplicidade de conceitos envolvidos justificam o uso de um sistema de informação para aplicação da MCDA-C. Um aplicativo computacional que implementa essa metodologia foi desenvolvido como resultado deste trabalho. Com o uso desse aplicativo cada gestor ou decisor pode definir o seu modelo de apoio à tomada de decisão. A ferramenta é flexível e permite personalizar os modelos de acordo com os critérios do decisor e as especificidades de cada problema a ser tratado ou de cada recurso a ser aplicado.

Palavras-chave: MCDA-C. Ferramenta de apoio à tomada de decisão. Ferramenta de gestão.

ABSTRACT

MOSER, Luiz Carlos. Software to implement Methodology Multicriteria Decision Aid – Constructivist. 2015. 57f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

Decision-making regarding the use of resources can be dependent on many criteria and variables. This process becomes more critical as resources are scarce. In public institutions such as Universities, the application of resources is always a difficult decision task. The multiplicity of interests and needs involved in applying resources makes this laborious and difficult process. Methodology MultiCriteria Decision Aid - Constructivist aims to provide a process of decision support based on interests and decision-making criteria. The complexity of the methodology and the multiplicity of concepts involved justify the use of an information system for application of the MCDA-C. A computer application that implements this methodology was developed as a result of this work. Using this application each manager or decision maker can set your decision making to support model. The tool is flexible and allows you to customize the models according to the criteria of the decision maker and the specifics of each problem being treated or each resource to be applied.

Palavras-chave: Methodology Multicriteria Decision Aid – Constructivist. Decision-making support tool. Management tool.

LISTA DE FIGURAS

Figura 1 – Fases da metodologia MCDA-C.....	18
Figura 2 – Exemplo da fase de estruturação da metodologia MCDA-C	19
Figura 3 – Exemplo da fase de avaliação da metodologia MCDA-C.....	20
Figura 4 – Exemplo da fase de recomendações da metodologia MCDA-C	21
Figura 5 – Diagrama de atividades do fluxo principal do sistema	26
Figura 6 – Diagrama de casos de uso.....	29
Figura 7 – Unidades conceituais do sistema	31
Figura 8 – Diagrama de sequência.....	32
Figura 9 – Diagrama de classes do sistema.....	33
Figura 10 – Diagrama de entidades e relacionamentos do banco de dados	34
Figura 11 – Tela principal do sistema.....	35
Figura 12 – Itens de menu com funções específicas.....	36
Figura 13 – Tela padrão de navegação.....	37
Figura 14 – Tela padrão de cadastros	38
Figura 15 – Tela de cadastro de contexto.....	39
Figura 16 – Tela de cadastro de EPAs e conceitos.....	40
Figura 17 – Editor da estrutura hierárquica.....	41
Figura 18 – Editor auxiliar do indicador da estrutura hierárquica.....	41
Figura 19 – Editor auxiliar dos conceitos do indicador da estrutura hierárquica.....	42
Figura 20 – Editor auxiliar das escalas dos descritores	42
Figura 21 – Editor da estrutura hierárquica expandida.....	43
Figura 22 – Cadastro do projeto	44
Figura 23 – Tela de seleção de projetos.....	44
Figura 24 – Visualização do mapa decisório.....	45

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação	22
Quadro 2 – Iterações definidas	23
Quadro 3 – Requisito cadastrar usuários.....	26
Quadro 4 – Requisito indicadores.....	27
Quadro 5 – Requisito cadastro de modelos.....	27
Quadro 6 – Requisito descritores.....	27
Quadro 7 – Requisitos suplementares	28
Quadro 8 – Casos de uso.....	28
Quadro 9 – Caso de uso cadastrar usuário	29
Quadro 10 – Caso de uso cadastrar modelo	30
Quadro 11 – Caso de uso cadastrar requisitos.....	30
Quadro 12 – Caso de uso descritores	30

LISTAGEM DE CÓDIGOS

Listagem 1 – Método ActionListUpdate.....	46
Listagem 2 – Método ActionListExecute	47
Listagem 3 – Código da declaração da classe padrão de cadastro	48
Listagem 4 – Métodos para ler e gravar dados nas tabelas de cadastro.....	49
Listagem 5 – Implementação da funcionalidade da grid de navegação de dados	49
Listagem 6 – Código do método editar	50
Listagem 7 – Código do método inserir.....	50
Listagem 8 – Código do método excluir	51
Listagem 9 – Código do método Preparar implementado pela classe Pessoa.....	51
Listagem 10 – Código do método Validar	52
Listagem 11 – Código do método Inserir.....	52
Listagem 12 – Código do método Editar	53
Listagem 13 – Código do método Gravar	53
Listagem 14 – Código canvas do indicador	54
Listagem 15 – Código canvas do descritor	54

LISTA DE ABREVIATURAS E SIGLAS

EPA	Elementos Primários de Avaliação
MCDA-C	Metodologia Multicritério de Apoio à Decisão – Construtivista
PVE	Ponto de Vista Elementar
PVF	Ponto de Vista Fundamental
WYSIWYG	<i>What You See Is What You Get</i>

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 CONSIDERAÇÕES INICIAIS	13
1.2 OBJETIVOS.....	14
1.2.1 Objetivo Geral.....	14
1.2.2 Objetivos Específicos.....	14
1.3 JUSTIFICATIVA	15
1.4 ESTRUTURA DO TRABALHO	15
2 REFERENCIAL TEÓRICO	16
2.1 APLICAÇÃO DE RECURSOS EM INSTITUIÇÕES DE ENSINO.....	16
2.2 METODOLOGIA MULTICRITÉRIO DE APOIO À DECISÃO CONSTRUTIVISTA ..	17
3 MATERIAIS E MÉTODO	22
3.1 MATERIAIS.....	22
3.2 MÉTODO	22
4 RESULTADOS	25
4.1 ESCOPO DO SISTEMA.....	25
4.2 MODELAGEM DO SISTEMA.....	26
4.3 APRESENTAÇÃO DO SISTEMA.....	35
4.4 IMPLEMENTAÇÃO DO SISTEMA	45
5 CONCLUSÃO.....	55
REFERÊNCIAS.....	56

1 INTRODUÇÃO

A seguir estão as considerações iniciais que apresentam o contexto deste trabalho, os seus objetivos e a justificativa de realização do mesmo. O capítulo é finalizado com a apresentação dos capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

As instituições privadas e públicas necessitam frequentemente investir recursos para que possam produzir, de modo eficiente, os resultados planejados para a atividade desempenhada. Esses recursos, na grande maioria das vezes, escassos, precisam ser aplicados de forma a maximizar o seu retorno e ao mesmo tempo atender as necessidades mais prioritárias da instituição.

Tais recursos geralmente são alocados seguindo a percepção de seus gestores com tomadas de decisão baseadas em dados insuficientes e coletados sem critérios específicos. A aplicação dos recursos sem uma adequada fundamentação para a decisão de aplicá-los pode causar desperdícios financeiros e insatisfação das pessoas.

Os recursos computacionais são vistos como ferramentas para auxiliar os gestores. O desenvolvimento de ferramentas para coleta de dados e auxílio na posterior análise desses dados fornece aos gestores a base necessária para tomada de decisões eficazes com a melhor eficiência de aplicação de recursos. Assim, questiona-se quais seriam os melhores requisitos para a construção de uma ferramenta para elaboração de modelos estruturados e confiáveis a fim de suprir as necessárias dos gestores na aplicação dos recursos da instituição?

Para que essas ferramentas possam ser desenvolvidas e utilizadas com efetividade é necessário que elas representem conceitos e modelos de negócio e que possibilitem o melhor uso do conhecimento dos gestores com o apoio de dados produzidos no ambiente institucional e externo.

Um desses modelos é a Metodologia Multicritério de Apoio à Decisão – Construtivista (MCDA-C) que visa atender a essas etapas do gerenciamento do capital intelectual (SCHNORRENBARGER, 2005). A MCDA-C é uma forma de auxiliar na tomada de decisão, considerando as percepções e os valores dos envolvidos no processo (ENSSLIN, 2002), sob a perspectiva de uma visão ampla do conhecimento sobre a tomada de decisões (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001).

O público alvo inicial do sistema de gestão desenvolvido utilizando a metodologia MCDA-C são instituições de ensino e o câmpus de Pato Branco da Universidade Tecnológica Federal do Paraná será utilizado para avaliar o aplicativo desenvolvido e como base para a coleta de dados. Contudo, esse modelo se aplica a qualquer instituição de ensino e mesmo organizações de negócio. Isso, porque a ferramenta possibilita que cada gestor construa o seu modelo de critérios para avaliação da aplicação de recursos.

1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos do trabalho.

1.2.1 Objetivo Geral

Desenvolver uma ferramenta de apoio à tomada de decisão que implementada a metodologia MCAD-C.

1.2.2 Objetivos Específicos

- Coletar requisitos para entender como o processo de tomada de decisões é realizado utilizando a metodologia MCDA-C.
- Levantar as dificuldades enfrentadas pelos gestores visando construir um modelo que atenda os interesses dos gestores e a metodologia MCDA-C.
- Apresentar mapas visuais com base nos dados coletados produzindo previsões de alocação de recursos.
- Possibilitar a análise dos resultados da aplicação de recursos qualificando a eficiência do modelo adotado nas decisões tomadas pelos gestores.

1.3 JUSTIFICATIVA

Justifica-se o desenvolvimento de um aplicativo computacional para organizar de forma sistemática os elementos necessários para que gestores de instituições públicas de ensino superior tenham como base um modelo gerenciável de elementos decisórios no processo de repasse dos recursos institucionais aos seus respectivos setores. O uso da MCDA-C fornece a fundamentação conceitual para definir o modelo de negócio a ser adotado como fundamentação para a tomada de decisão.

Atualmente os gestores dependem de ferramentas fragmentadas e conhecimento próprio para elaboração de planilhas, gráficos e documentos que servem como base de argumentação para justificar o pedido do repasse de determinado recurso, quer seja recursos humanos, quer seja recursos financeiros. Assim, as decisões de quais recursos devem ser disponibilizados e quais os setores serão contemplados ficam a cargo de debates das comissões formadas pelos interessados, os gestores que dispõe dos recursos e os que reivindicam tais recursos.

Sendo assim, a proposta de construção de uma ferramenta de software capaz de automatizar o processo, por meio da elaboração de projetos prévios embasados em dados, contribui para a tomada consciente de decisões pelos gestores. Aumentando, dessa forma, a eficiência na alocação dos recursos produzindo, como resultado final, dados comparativos necessários para a medição da eficácia no aproveitamento de tais recursos.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos. O Capítulo 2 apresenta o referencial teórico que abrange a aplicação de recursos em instituições de ensino e a metodologia multicritério de apoio à decisão construtivista. No Capítulo 3 são apresentadas as ferramentas e as tecnologias utilizadas na modelagem e desenvolvimento do aplicativo e o método. O método é composto pelas principais etapas e respectivas atividades realizadas. O resultado obtido até o momento está apresentado no Capítulo 4. O trabalho é finalizado com as considerações finais seguidas das referências utilizadas no texto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico que fundamenta o sistema desenvolvido como resultado deste trabalho. O referencial teórico se refere à aplicação de recursos e a metodologia multicritério de avaliação.

2.1 APLICAÇÃO DE RECURSOS EM INSTITUIÇÕES DE ENSINO

Fraidenraich (2010, p.1) ressalta que “administrar os recursos financeiros de uma escola não é tarefa fácil. É preciso avaliar muito bem onde aplicá-los de forma que tenham reflexos na qualidade do ensino e na aprendizagem dos alunos.”

O entendimento do processo de gestão de recursos, seja em instituições privadas ou públicas, está vinculado à forma de tomada de decisão para aplicação dos recursos. E nem sempre os recursos estão acessíveis aos setores interessados ou que efetivamente necessitam dos recursos.

Para que a alocação dos recursos seja feita de maneira adequada visando o atendimento de necessidades prioritárias e maximizando os seus resultados é importante que haja mecanismos claros, de conhecimento de todos e consistentes.

A prática de que alguém decide e todo mundo faz está ultrapassada e não condiz com o conceito de autonomia da escola. Por isso, o ideal é promover reuniões periódicas com representantes dos diversos segmentos - alunos, professores, funcionários, pais e responsáveis - durante o ano para que todos tenham informações sobre as necessidades da instituição, ajudem a elencar as prioridades e acompanhem a execução dos recursos (FRAIDENRAICH, 2010, p.1).

Com a participação de diversos segmentos no processo de aplicação dos recursos pode-se maximizar os investimentos em áreas de interesse da comunidade e uma melhor avaliação das necessidades. A necessidade de investimento deve seguir critérios no sentido evolutivo dando melhores condições para o aprendizado, conforme o PPP (Projeto Político Pedagógico), considerando a capacitação dos professores e o ambiente de ensino.

Fraidenraich (2010, p.1) ressalta que “é fundamental mostrar de que forma aquele recurso impactou a aprendizagem do aluno.” Isso no sentido da aplicação dos recursos que a instituição teve como entrada e indicar a saída ou resultado obtido da aplicação do referido recurso. Ferramentas computacionais podem ser aplicadas para a construção de índices

comparativos para auxiliar os gestores no gerenciamento da aplicação de recursos e tomada de decisão.

2.2 METODOLOGIA MULTICRITÉRIO DE APOIO À DECISÃO CONSTRUTIVISTA

A origem da Metodologia Multicritério de Apoio à Decisão – Construtivista (MCDA-C), segundo Ensslin *et al.* (2010) tem suas origens há mais de dois séculos. Contudo, esses autores ressaltam que a sua consolidação como instrumento de gestão é da década de 1980. E que isso ocorreu com os trabalhos de Roy (1996) e Landry (1995) ao definir os limites da objetividade para os processos de apoio à decisão, de Skinner (1986) e Keeney (1992) ao reconhecer que os atributos (objetivos, critérios) são específicos ao decisor em cada contexto e de Bana e Costa (1993) ao explicitar as convicções da MCDA.

A MCDA-C surge como uma ramificação da MCDA tradicional para apoiar os decisores em contextos complexos, conflituosos e incertos (ENSSLIN *et al.*, 2010, p. 128). A multiplicidade de variáveis qualitativas e quantitativas define o caráter de complexidade. O envolvimento de múltiplos atores, que podem possuir interesses distintos e não alinhados com as preocupações do decisor, define o caráter de conflito. E incertos estão relacionados a não certeza das decisões tomadas diante de ocorrências internas e externas quando há disputa pela aplicação de recursos que geralmente são escassos e as necessidades também são muitas e diversificadas.

A MCDA-C é uma forma de auxiliar na tomada de decisão, considerando as percepções e os valores dos envolvidos no processo (ENSSLIN, 2002), na perspectiva de uma visão global sobre os fatores e o contexto que envolve a tomada de decisão (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001). Por meio dessa metodologia, o conhecimento é construído ou gerado a partir do contexto ou situação decisional, obtido pela interação entre o sujeito e o objeto.

Visando cumprir o objetivo proposto com o uso da MCDA – C no processo de tomada de decisão são utilizadas atividades de apoio à decisão, subdivididas em fases (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001). A operacionalização dos fundamentos da metodologia MCDA-C é realizada por meio das três fases básicas: estruturação do contexto decisório, construção de um modelo de avaliação de alternativas e/ou ações e formulação de recomendações para os cursos de ações mais satisfatórios. Essas fases são apresentadas na Figura 1.

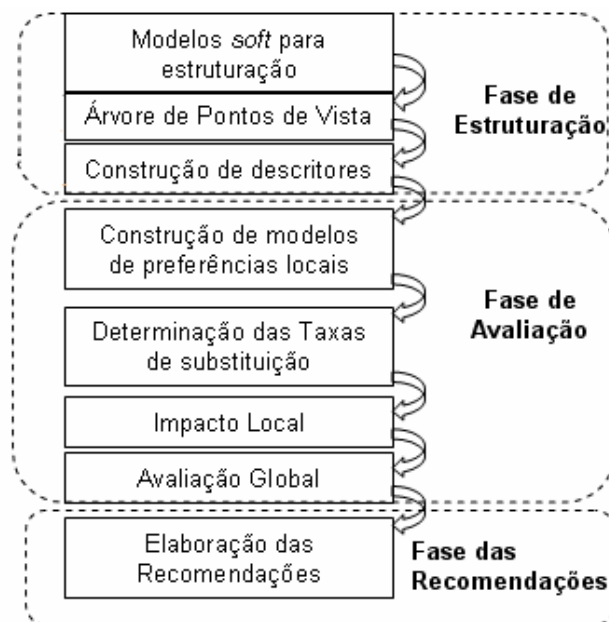


Figura 1 – Fases da metodologia MCDA-C

Fonte: Adaptado de Ensslin, Dutra e Ensslin (2000) *apud* Ensslin et al. (2010, p. 131).

a) Fase de estruturação – entendimento e estruturação do contexto envolvido no processo decisório. A identificação dos atores envolvidos e a identificação dos pontos de vista e conceitos envolvidos é a etapa que segue a contextualização.

Os atores são agentes que de alguma forma estão envolvidos no processo decisório por meio do seu sistema de valores, expressando suas preferências, com o propósito de atingir os seus objetivos (ZANELLA, 1996). Cada ator é influenciado pelo sistema de valores dos demais atores e todos interagem com o ambiente e contexto do problema sendo analisado e da solução que será proposta. Na fase de contextualização é definido o rótulo do problema para o qual é buscada uma solução (TEZZA; ZAMCOPÉ; ENSSLIN, 2010).

O levantamento dos Elementos Primários de Avaliação (EPAs) é realizado na fase de estruturação, após o problema e os atores serem definidos. Os EPAs geram o entendimento e a definição dos limites entre o pretendido e o mínimo aceitável em termos de solução do problema (TEZZA; ZAMCOPÉ; ENSSLIN, 2010). Os EPAs são organizados em mapas cognitivos. Esses mapas, para Ensslin, Montibeller Neto e Noronha (2001), representam uma hierarquia de conceitos relacionados por ligações de influência entre meios e fins. O mapa cognitivo gera uma árvore de pontos de vista ou estrutura hierárquica de valores (KEENEY, 1992). Os pontos de menor complexidade são conectados aos pontos de maior complexidade possibilitando mensurá-los. Os pontos de vista são classificados em: Ponto de Vista Fundamental (PVF) que representam os aspectos essenciais, de acordo com o decisor, que

definem os eixos de avaliação do problema; e Pontos de Vista Elementar (PVE) que são os aspectos meio para atingir um fim, ou seja, meios para alcançar os PVF (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001).

Da identificação de todos os PVEs são obtidos os descritores. A construção dos descritores está envolvida na identificação de um conjunto de níveis de impacto que representam, de forma não ambígua, os possíveis desempenhos de uma ação, ordenados de acordo com o sistema de valor adotados pelos decisores que estão envolvidos no processo (BANA E COSTA; SILVA, 1994).

A Figura 2 apresenta um exemplo do modelo para a fase de estruturação. Esse modelo é de apoio à decisão para a distribuição de vagas docentes.

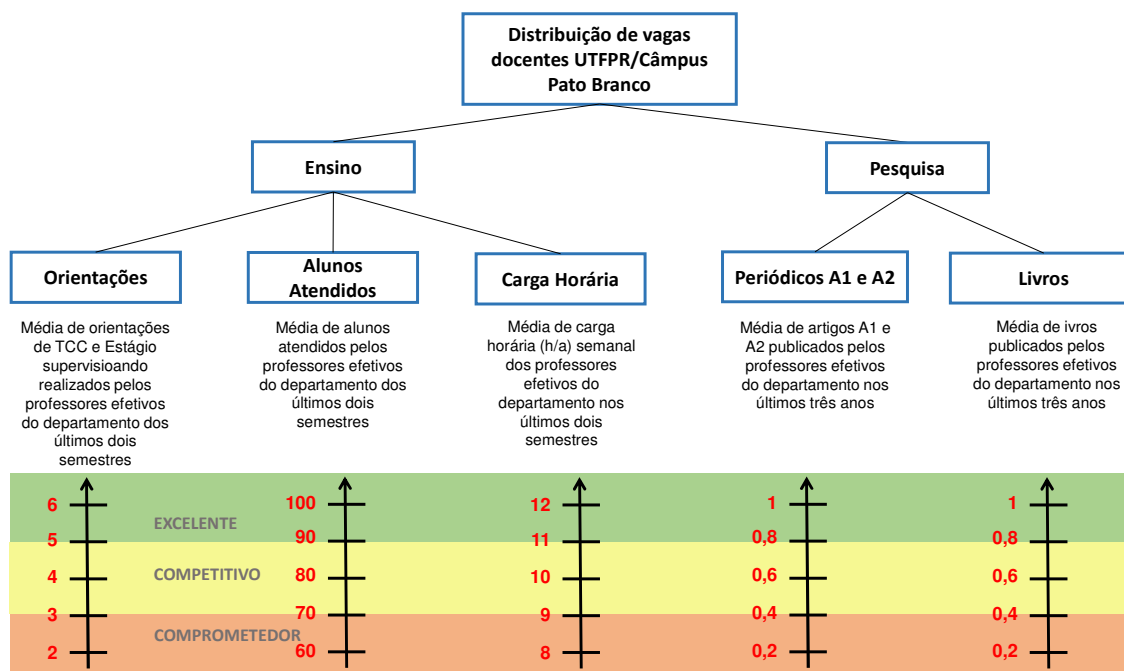


Figura 2 – Exemplo da fase de estruturação da metodologia MCDA-C

Fonte: autoria própria.

b) Fase de avaliação – nessa fase são construídas as funções de valor para cada descritor. Uma função de valor é a representação matemática da intensidade de preferência entre os níveis de impacto de um descritor (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001).

Nesse contexto os valores são apresentados na escala, os ordinais representando valor quantitativos ou qualitativos e os valores cardinais representando a contraposição dos valores ordinais. Os valores de contraposição de ordinais podem ser obtidos com o uso da ferramenta

MacBeth (MACBETH, 2015). Na Figura 3 está o modelo gerado na fase de avaliação para o exemplo de distribuição de vagas docentes.

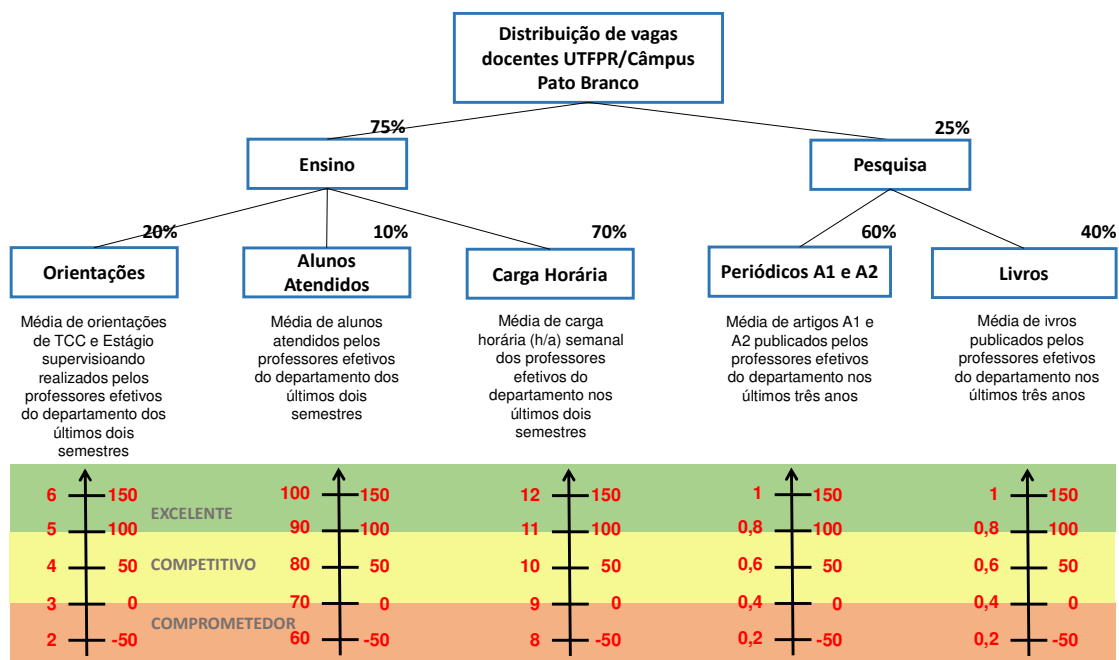


Figura 3 – Exemplo da fase de avaliação da metodologia MCDA-C

Fonte: autoria própria.

c) Fase de elaboração das recomendações – as recomendações se referem às ações de melhoria e aperfeiçoamento dos critérios. Essas ações são definidas de acordo com o problema que está sendo analisado. Esse problema pode se referir a aplicação de determinado recurso. A análise da sensibilidade das respostas em relação às variações dos parâmetros utilizados com os dados de entrada, a geração de estratégias orientadas à ação e a análise de sensibilidade do modelo são realizados considerando o desempenho das ações (ENSSLIN; MONTIBELLER NETO; NORONHA, 2001).

A fase de elaboração de recomendações, do modelo de apoio à decisão para a distribuição de vagas docentes, é apresentada na Figura 4.

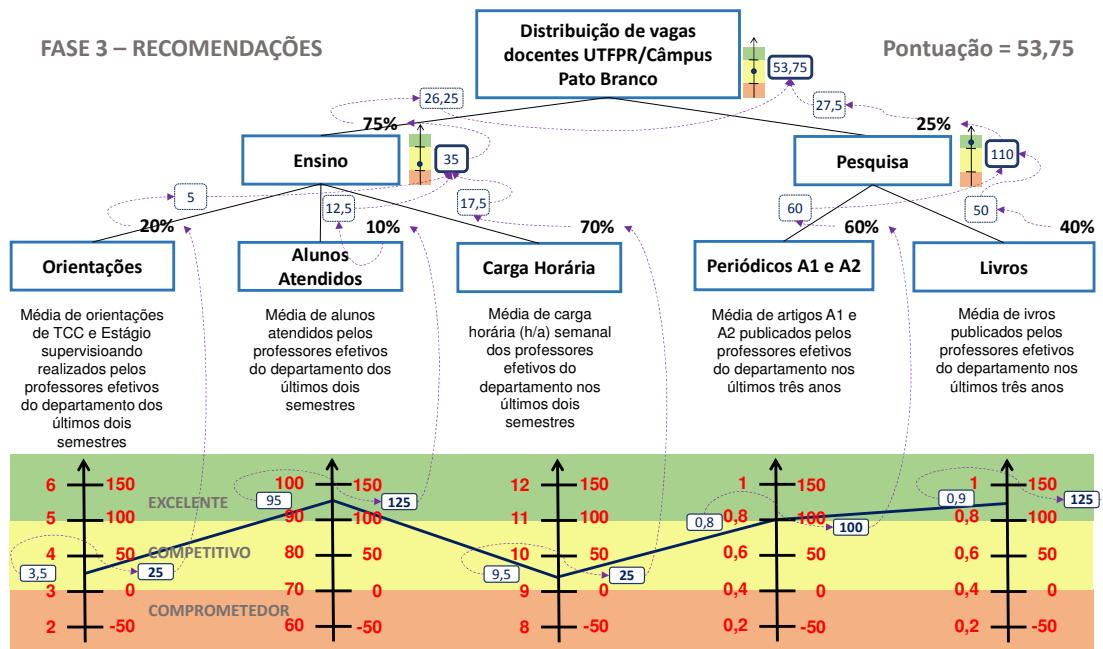


Figura 4 – Exemplo da fase de recomendações da metodologia MCDA-C
 Fonte: autoria própria.

3 MATERIAIS E MÉTODO

A ênfase deste capítulo está em reportar o que e como será feito para alcançar o objetivo do trabalho. Este capítulo está subdividido, em duas seções, sendo uma para os materiais e outra para o método.

3.1 MATERIAIS

Para a modelagem e a implementação do sistema foram utilizadas as ferramentas e as tecnologias apresentadas no Quadro 1.

Ferramenta / Tecnologia	Versão	Finalidade
Visual Paradigm	10.1/10.2 Community	Para a modelagem do aplicativo e da base de dados.
Delphi	XE5	Linguagem de programação e ambiente de desenvolvimento.
SQLite	3.0	Banco de dados embarcado.

Quadro 1 – Tecnologias e ferramentas utilizadas na modelagem e na implementação

3.2 MÉTODO

A modelagem e o projeto de desenvolvimento do aplicativo que implementa a metodologia multicritério para apoio à tomada de decisão elaborados têm como base o modelo sequencial linear (PRESSMAN, 2008) para a definição dos processos e o processo unificado (BLAHA *et al.*, 2006) para a definição das iterações.

O Quadro 2 apresenta os processos (fluxos de trabalho) e as iterações desenvolvidas. Essas iterações complementam as atividades realizadas como trabalho de estágio.

Iterações / Processos	1ª iteração	2ª iteração	3ª iteração
Requisitos	Definição dos requisitos. Resultado: listagem dos requisitos	Complementos à definição dos requisitos Resultado: ajustes na listagem dos requisitos	Complementos à definição dos requisitos Resultado: ajustes na listagem dos requisitos
Análise e projeto	Modelagem dos requisitos Resultado: diagrama de atividade do fluxo principal do sistema, diagrama de casos de uso, classes e	Modelagem dos requisitos Resultado: diagrama do banco de dados ajustado, tabelas com campos	Modelagem dos requisitos Resultado: diagrama de classes complementado para

	entidades do banco de dados.	incluídos.	atender os ajustes realizados no banco de dados.
Implementação	Dos cadastros. Resultado: cadastros básicos do sistema	Da composição da fase de estruturação. Resultado: do editor para a composição dos diagramas	Da fase de avaliação Resultado: editor com permissão de editar os conceitos
Testes	De código, realizados pelo programador (o autor do trabalho). De requisitos realizados pelo especialista. De interação e interface realizados pela orientadora e especialista Resultado: avaliação do sistema realizada pelo especialista, visando validar a interação com o sistema.	De código, realizados pelo programador (o autor do trabalho). De requisitos realizados pelo especialista. De interação e interface realizados pela orientadora e especialista. Resultado: avaliação do sistema realizada pelo especialista, visando validar o editor.	De código, realizados pelo programador (o autor do trabalho). De requisitos realizados pelo especialista. De interação e interface realizados pela orientadora e especialista. Resultado: avaliação do sistema realizada pelo especialista, visando validar a edição dos conceitos no editor.

Quadro 2 – Iterações definidas

A seguir estão descritas as principais atividades realizadas nos processos constantes no Quadro 2.

a) Levantamento de requisitos

O levantamento dos requisitos foi realizado a partir de conversas com especialista na metodologia MCDA-C. Esse especialista forneceu a visão geral da metodologia e os requisitos que o sistema deveria ter. A partir dessa visão geral foram definidos os requisitos funcionais e não funcionais, bem como os tipos de usuários e as permissões que os mesmos teriam no sistema.

b) Análise e projeto do sistema

A análise e projeto consistiram na modelagem dos requisitos do sistema por meio de diagramas e protótipo das telas.

c) Implementação

A implementação foi realizada à medida que os requisitos eram definidos. Na primeira iteração foram implementados os cadastros básicos do sistema. Na segunda iteração, os requisitos que permitiam realizar a fase de estruturação da metodologia foram implementados. Na terceira iteração os requisitos que permitiam implementar a fase de avaliação.

d) Testes

Os testes de código foram realizados pelo autor deste trabalho. Os testes de requisitos e interação foram realizados por especialista na metodologia MCDA-C. Os testes de interface (interação com o sistema) foram realizados pela orientadora e por especialistas.

4 RESULTADOS

Este capítulo apresenta o resultado deste trabalho que é o desenvolvimento de uma ferramenta para aplicação da metodologia MCDA-C. No capítulo também constam códigos que visam demonstrar como a implementação foi realizada com o objetivo do estudo e conhecimento da metodologia utilizada. A implementação será efetivada como trabalho de conclusão de curso.

4.1 ESCOPO DO SISTEMA

O sistema modelado como resultado deste trabalho automatizará o procedimento de avaliação de critérios para tomada de decisão dos gestores. Como exemplo, está a distribuição de recursos aos setores acadêmicos, as bolsas destinadas aos projetos acadêmicos e a avaliação de desempenho acadêmico. A solução proposta considera o contexto apresentado a seguir.

Um projeto de avaliação é construído na forma de estrutura hierárquica por um profissional da área de administração ou outro profissional com o conhecimento da metodologia. Com base na entrevista com o gestor, o profissional busca elencar os pontos de avaliação denominados de conceitos. Um conceito é um item de avaliação que pode ser distribuído em indicadores de desempenho com a finalidade de avaliação mais detalhada em pontos específicos onde cada ponto é denominado de descritor. Neste caso, o descritor contém uma escala com valores ordinais e cardinais. Os valores ordinais podem ser quantitativos ou qualitativos.

O descritor é um elemento decisório que questiona o gestor tendo como objetivo produzir os dados necessários ao cálculo da pontuação dos critérios. Esses dados inseridos na ferramenta servem de base para a construção do modelo proposto. Como resultado final temos a análise de desempenho para comparativo entre as diversas áreas, ou setores, diretamente relacionados com a distribuição do recurso disponibilizado pelo gestor.

O sistema possibilitará visualizar um mapa gráfico com a distribuição dos requisitos selecionados mostrando três faixas sendo “Excelente”, “Competitivo” e “Comprometedor”. O ideal é que os valores dos requisitos fiquem na faixa “Competitivo” indicando normalidade, enquanto que a faixa “Comprometedor” indica deficiência e a faixa “Excelente” indicando desempenho acima da normalidade.

Após a estrutura hierárquica ou árvore de requisitos que define um projeto, ser composta, é possível alimentar os descritores. Os descritores são formados pelos dados coletados para cada requisito em questão. Esses descritores serão utilizados para a produção dos mapas gráficos com suas respectivas escalas quantitativas e qualitativas distribuídos de forma linear horizontalmente com as três faixas de cores conforme a escala de cada descritor.

4.2 MODELAGEM DO SISTEMA

O diagrama de atividades que apresenta o fluxo principal do sistema está na Figura 5.

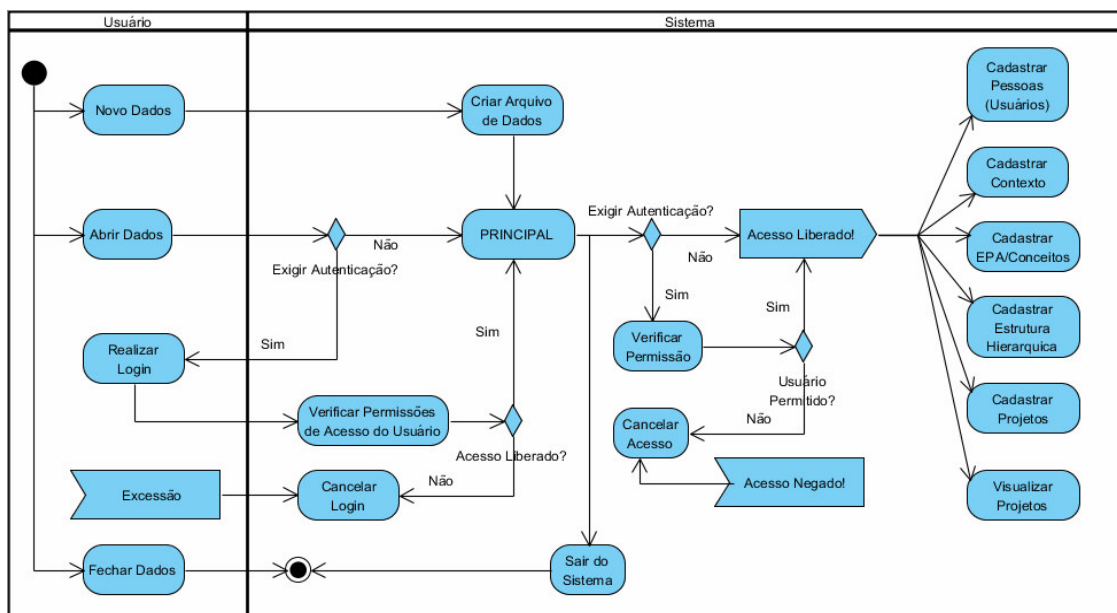


Figura 5 – Diagrama de atividades do fluxo principal do sistema

A seguir estão apresentados os principais requisitos funcionais e não funcionais definidos para o sistema. No Quadro 3 está o requisito cadastrar usuários.

Nome		Restrição	Categoria	Desejável	Permanente
NF1.1 Níveis de acesso por usuário		O usuário deve conter o nível de acesso aos dados cadastrados.	Hardware	(x)	(x)

Quadro 3 – Requisito cadastrar usuários

O requisito funcional para o cadastro de indicadores e os respectivos requisitos não funcionais está representado no Quadro 4.

F2 Indicadores		Oculto ()		
Descrição: O sistema deve oferecer ao usuário administrador a possibilidade de cadastrar os indicadores. Além disso, deve ser permitido editar e excluir cadastros.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Visualização de Indicadores	Os indicadores devem ser apresentados em forma de árvore.	Interface	()	(x)
NF2.1 Substituição de Valores	Possibilidade de substituir os valores de um indicador durante a alimentação do seu conjunto de amostragem.	Sistema	()	(x)
NF2.3 Conceitos	Permitir distribuir no indicador o conceito cadastrado na EPA, apenas para nível 1 e 2 da árvore.	Sistema	()	(x)

Quadro 4 – Requisito indicadores

No Quadro 5 está o requisito funcional cadastro de modelos e os requisitos não funcionais associados.

F3 Cadastro Modelos		Oculto ()		
Descrição: O sistema conter um cadastro de modelos (mapas relacionais), a cada modelo cadastrado deve conter um ou muitos indicadores qualitativos e quantitativos.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Gera mapa	Os modelos devem ter a possibilidade de gerar mapas comparativos entre os indicadores cadastrados.	Interface	()	(x)
NF3.2 Visualização	Os mapas devem ser apresentados em forma gráfica linear, separados por cores diferentes em relação aos valores abaixo do mínimo, acima do máximo e mediana.	Sistema	(x)	(x)

Quadro 5 – Requisito cadastro de modelos

O requisito funcional cadastrar descritores e os requisitos não funcionais associados estão representados no Quadro 6.

F4 Descritores		Oculto ()		
Descrição: O sistema deve possibilitar a inserção de descritores para a alimentação do conjunto de amostragem dos indicadores.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Visualização	Os descritores disponibilizam um prévia do resultado final.	Interface	()	(x)

Quadro 6 – Requisito descritores

Os requisitos suplementares são apresentados no Quadro 7.

Nome	Restrição	Categoria	Desejável	Permanente
S1	O sistema deve possuir uma interface que permita a manipulação dos objetos visuais em tela, contendo os índices e escalas, no formato hierárquico.	Usabilidade	()	(x)
S2	O sistema deve possuir um nível especial de acesso aos dados, de acordo com o tipo de usuário cadastrado.	Segurança	()	(x)
S3	O manual precisa ser uma referência completa do sistema com o qual o usuário possa aprender todas as funcionalidades, bem como os elementos de interface.	Documentação	(x)	(x)
S4	O sistema deve oferecer a opção que o usuário salve o backup do banco em um dispositivo que o ache conveniente.	Confiabilidade	(x)	(x)

Quadro 7 – Requisitos suplementares

O Quadro 8 apresenta os requisitos funcionais levantados para o sistema e o seu relacionamento com os casos de uso listados nos Quadros 3 a 6.

ID	Nome	Atores	Referências Cruzadas	Prioridade de realização
1	Cadastrar Usuários (CRUD)	Usuários	F1	1
2	Indicadores	Usuários, Entrevistadores	F1,F2	2
3	Cadastro Modelo	Usuários	F2,F3	1
4	Descritores	Usuários, Entrevistadores	F4,F2	2

Quadro 8 – Casos de uso

A Figura 6 apresenta o diagrama de casos de uso.

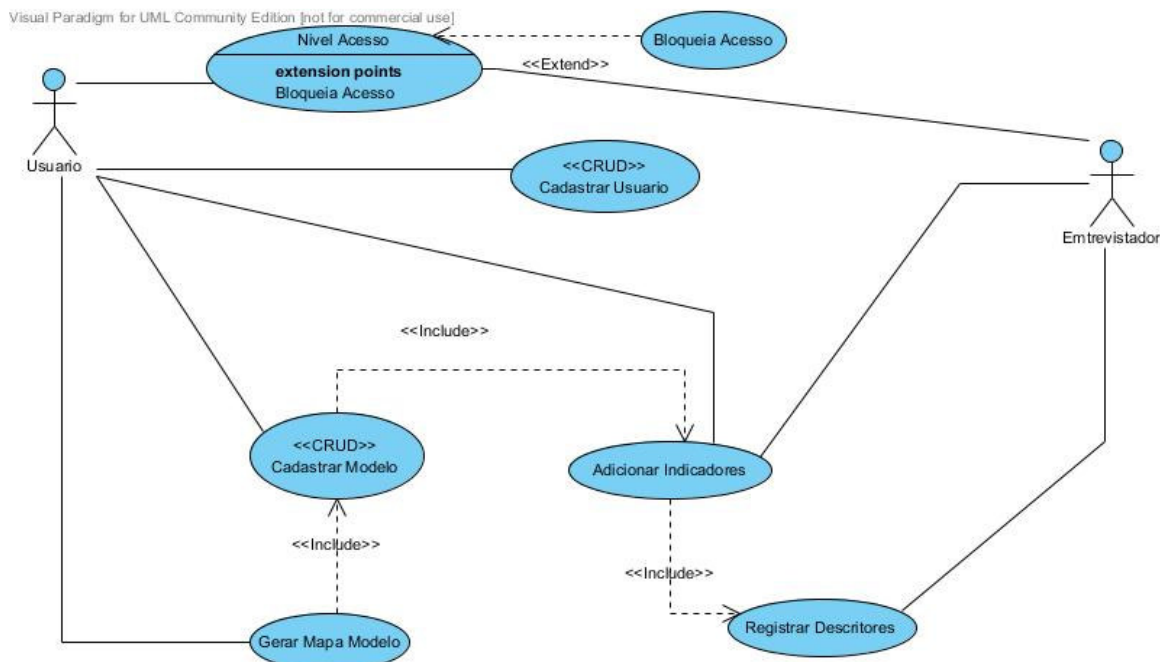


Figura 6 – Diagrama de casos de uso

A expansão dos casos de uso é apresentada nos Quadros 9 a 12. O Quadro 9 apresenta a expansão do caso de uso cadastrar usuário.

Caso de Uso: Cadastrar Usuário
Atores: Usuário
Precondições: O Usuário solicitará cadastro, caso não esteja cadastrado.
Pós-condições: Se o usuário não tiver permissão para cadastrar, o cadastro será cancelado.
Sequência típica de eventos (Fluxo Principal): Esse caso de uso inicia quando: <ol style="list-style-type: none"> 1. [IN] O usuário solicita o login e informando seu login 2. [OUT] O sistema valida o login do usuário. 3. [OUT] O sistema verifica se o usuário já está cadastrado. 4. [OUT] O sistema confirma o login. 5. [OUT] O sistema valida se o usuário tem acesso a cadastrar novos usuários. 6. [OUT] O sistema pede as informações para o cadastro do novo usuário.
Tratamento de Exceções e Variantes: Exceção 1a: Dados cadastrais do usuário inválidos ou inexistentes. <ol style="list-style-type: none"> 1a.1 [IN] O usuário informa os dados do novo usuário. 1a.2 [OUT] O sistema confirma o cadastro e retorna ao passo 1

Quadro 9 – Caso de uso cadastrar usuário

O caso de uso cadastrar modelo é apresentado no Quadro 10.

Caso de Uso: Cadastrar Modelo
Atores: Usuário
Precondições: O Usuário solicita o cadastro do modelo.
Pós-condições: Se o usuário não tiver permissão para cadastrar, o cadastro será cancelado.
Sequência típica de eventos (Fluxo Principal): Esse caso de uso inicia quando: <ol style="list-style-type: none"> 1. [IN] O usuário solicita o cadastro de modelo

<p>2. [OUT] O sistema valida se o usuário tem acesso a cadastrar novos modelos.</p> <p>3. [OUT] O sistema gera o modelo.</p> <p>4. [IN] O usuário informa os requisitos a serem vinculados com o modelo.</p>
<p>Tratamento de Exceções e Variantes: Exceção 1a: Dados cadastrais do modelo inválidos ou inexistentes. 1a.1 [IN] O usuário informa os requisitos a serem vinculados. 1a.2 [OUT] O sistema valida e confirma os requisitos a serem vinculados.</p>

Quadro 10 – Caso de uso cadastrar modelo

No Quadro 11 está a expansão do caso de uso requisitos.

Caso de Uso: Requisitos
Atores: Usuário, Entrevistador
Precondições: O Usuário os informa os requisitos coletados pelo entrevistador.
Pós-condições: Se o usuário não tiver permissão para inserir os dados do entrevistador, a inserção será cancelada.
<p>Sequência típica de eventos (Fluxo Principal): Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> 1. [IN] O usuário coleta os dados do entrevistador 2. [IN] O usuário solicita a inserção dos requisitos (dados). 3. [OUT] O sistema valida se o usuário tem acesso à inserção de requisitos. 4. [IN] O usuário informa os requisitos a serem inseridos. 5. [IN] O usuário vincula ao modelo os requisitos inseridos.
<p>Tratamento de Exceções e Variantes: Exceção 1a: Dados dos requisitos inseridos inválidos ou inexistentes. 1a.1 [IN] O usuário informa os requisitos a serem vinculados. 1a.2 [OUT] O sistema valida e confirma os requisitos a serem vinculados estão corretos e valida o vínculo com o modelo.</p>

Quadro 11 – Caso de uso cadastrar requisitos

O caso de uso descritores está no Quadro 12.

Caso de Uso: Descritores
Atores: Usuário, Entrevistador
Precondições: O Usuário os informa os descritores coletados pelo entrevistador.
Pós-condições: Se o usuário não tiver permissão para inserir os descritores coletados pelo entrevistador, a inserção será cancelada.
<p>Sequência típica de eventos (Fluxo Principal): Esse caso de uso inicia quando:</p> <ol style="list-style-type: none"> 1. [IN] O usuário coleta os descritores do entrevistador 2. [IN] O usuário solicita a inserção dos descritores. 3. [OUT] O sistema valida se o usuário tem acesso à inserção de descritores. 4. [IN] O usuário informa os descritores e o vínculo aos requisitos. 5. [IN] O usuário vincula os descritores aos requisitos já cadastrados inseridos.
<p>Tratamento de Exceções e Variantes: Exceção 1a: Dados dos descritores inseridos inválidos ou inexistentes. 1a.1 [IN] O usuário informa os descritores e o vínculo. 1a.2 [OUT] O sistema valida e confirma os descritores inseridos a serem vinculados estão corretos e valida o vínculo ao requisito.</p>

Quadro 12 – Caso de uso descritores

O diagrama com a representação das unidades conceituais do sistema é apresentado na Figura 7.

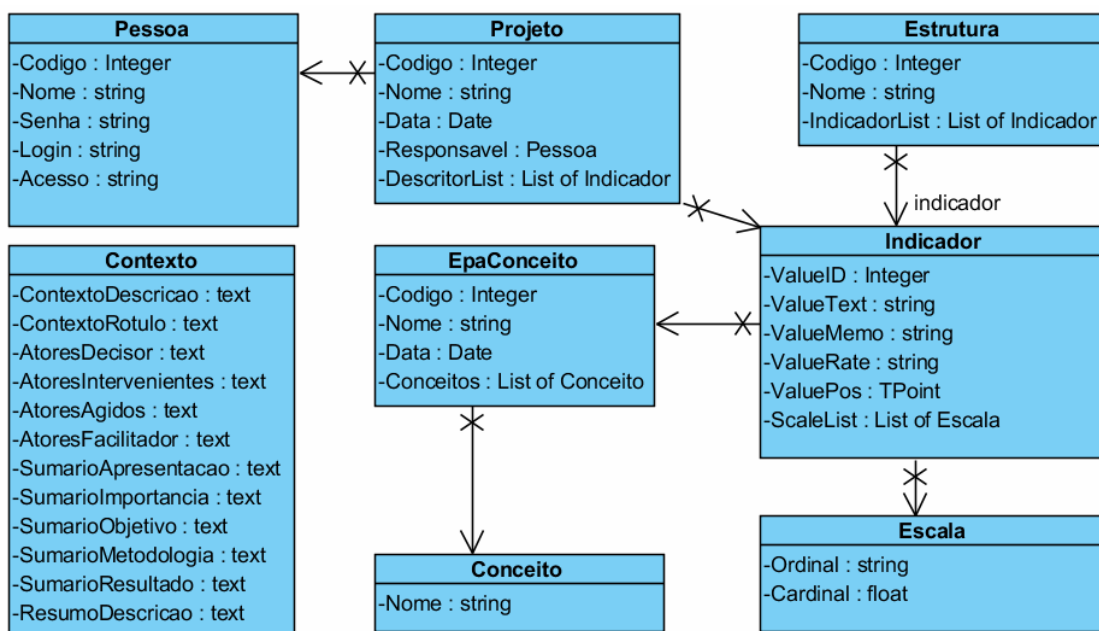


Figura 7 – Unidades conceituais do sistema

O diagrama de interação da funcionalidade principal do sistema é apresentado na Figura 8. Esse diagrama de interação contém três seqüências de interação com o usuário. Os itens 1, 2 e 3 são seqüências separadas no mesmo diagrama. E, assim, no sistema existem três tempos de processos.

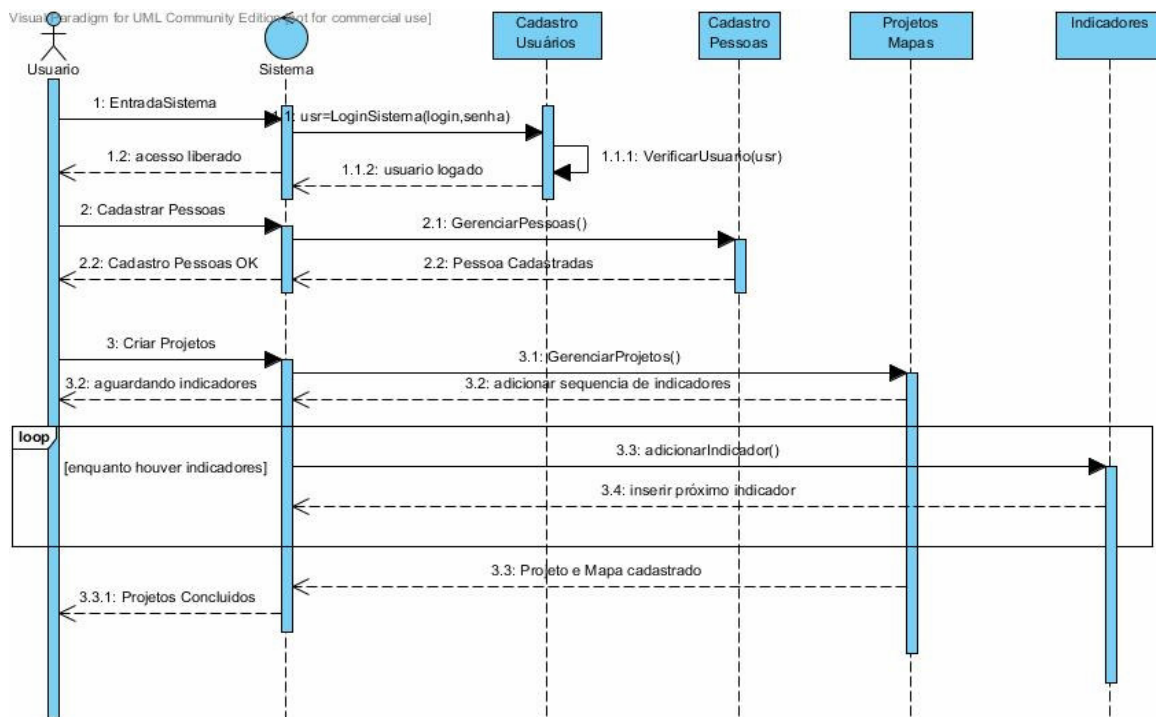


Figura 8 – Diagrama de seqüência

O Diagrama de Classes com atributos e métodos principais representa alguns dos elementos mais significativos do sistema, que são as classes visuais do sistema, é apresentado na Figura 9.

A Figura 10 apresenta o diagrama de entidades e relacionamentos do banco de dados.

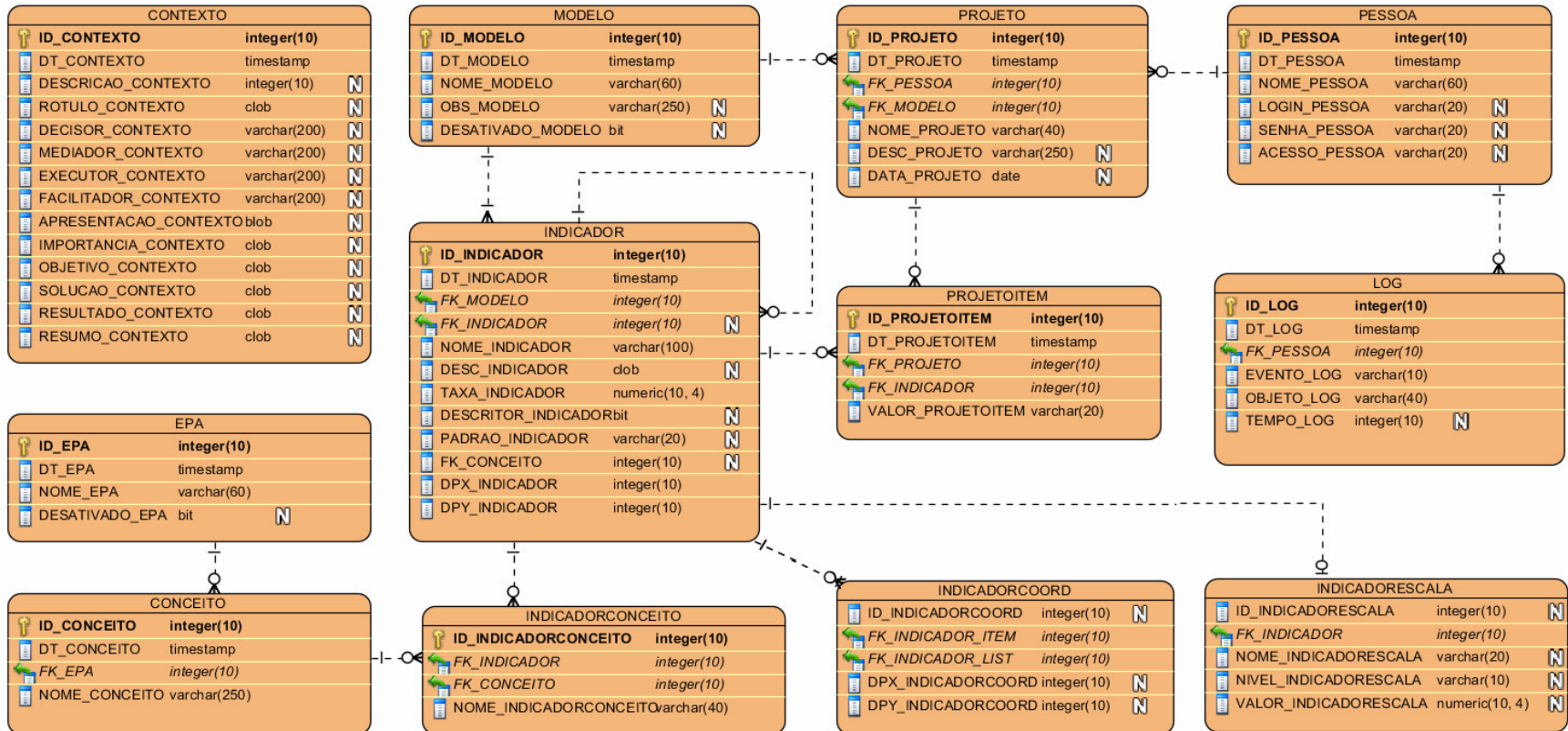


Figura 10 – Diagrama de entidades e relacionamentos do banco de dados

4.3 APRESENTAÇÃO DO SISTEMA

O leiaute do sistema é composto por um setor principal, a tela do sistema, contendo no topo um menu principal para chamada das opções do sistema. Além disso, há uma barra de botões logo abaixo do menu principal. Ambos, botões da barra de topo e menus, são ligados por componentes de ação, denominados *actions*, que controlam o acesso e a visibilidade das opções ao usuário. A tela principal é apresentada na Figura 11.

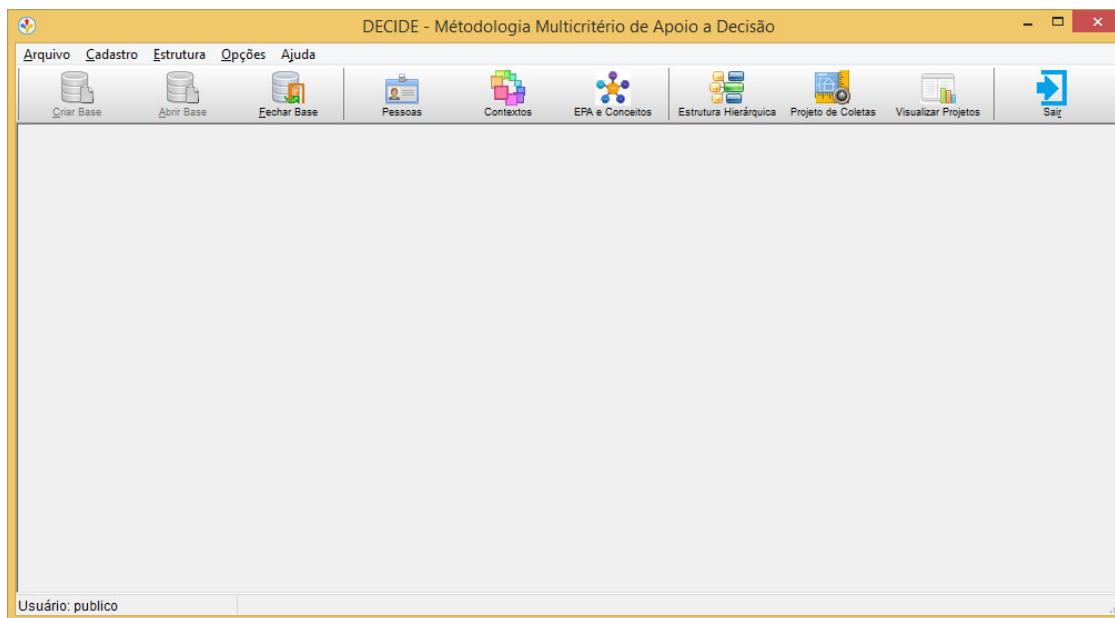


Figura 11 – Tela principal do sistema

Ainda na tela principal existem alguns elementos de configuração do sistema. No menu principal Arquivo está o item de menu Exigir Autenticação que indica ao sistema se deve solicitar o login de usuário sempre que abrir uma base de dados. Ao criar a base de dados essa opção encontra-se desmarcada e somente o usuário Administrador pode marcá-la ou desmarcá-la.

No menu principal Opções há dois itens de menu. O primeiro é o item Mostrar Valor Decisor, quando marcado, indica ao editor da estrutura hierárquica para mostrar os valores ordinais e cardinais encontrados junto com a linha gráfica de desempenho. O segundo é o item Mostrar Valor Aleatório, quando marcado, indica ao editor da estrutura hierárquica para produzir valores aleatórios durante a fase de estruturação para simular como ficaria o mapa dos descritores após o cadastramento das escalas. Na Figura 12 é apresentada os itens de menu com funções específicas.

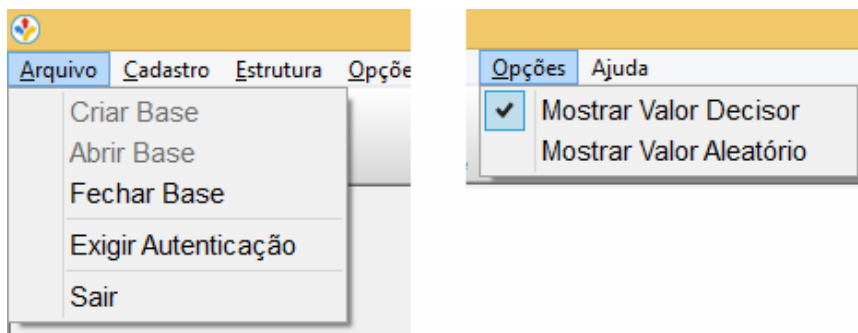


Figura 12 – Itens de menu com funções específicas

O sistema é construído com padronização de telas de cadastros contendo navegação de dados, inserção, edição e exclusão de registros. As telas de cadastros controlam quais os dados a tela de navegação deve exibir. Ao acessar uma tela de navegação de dados, por meio do menu principal ou barra de botões da janela principal do sistema, o sistema informa ao navegador de dados qual é a classe de cadastro responsável pelo controle das ações. O navegador de dados solicita quais dados devem ser exibidos e o título da janela, entre outras características para a classe de cadastro e armazena uma ação do usuário.

Quando o usuário acionar uma das ações da barra de rodapé do navegador de dados, este envia uma chamada respectiva para a classe de cadastro que é a responsável por atender o pedido. Ao acionar o botão “Inserir”, o navegador de dados solicita a ação de novo registro para a classe de cadastro passando como parâmetro o id com valor zero. Da mesma forma, a ação de “Alterar” faz a mesma solicitação para a classe de cadastro, desta vez, informando o id selecionado na tela do navegador de dados. Ambas as ações desempenham o mesmo papel, exceto pelo valor do id passado por parâmetro e a forma como os dados são carregados. Na Figura 13 está a tela padrão de navegação de dados utilizadas pelos cadastros.

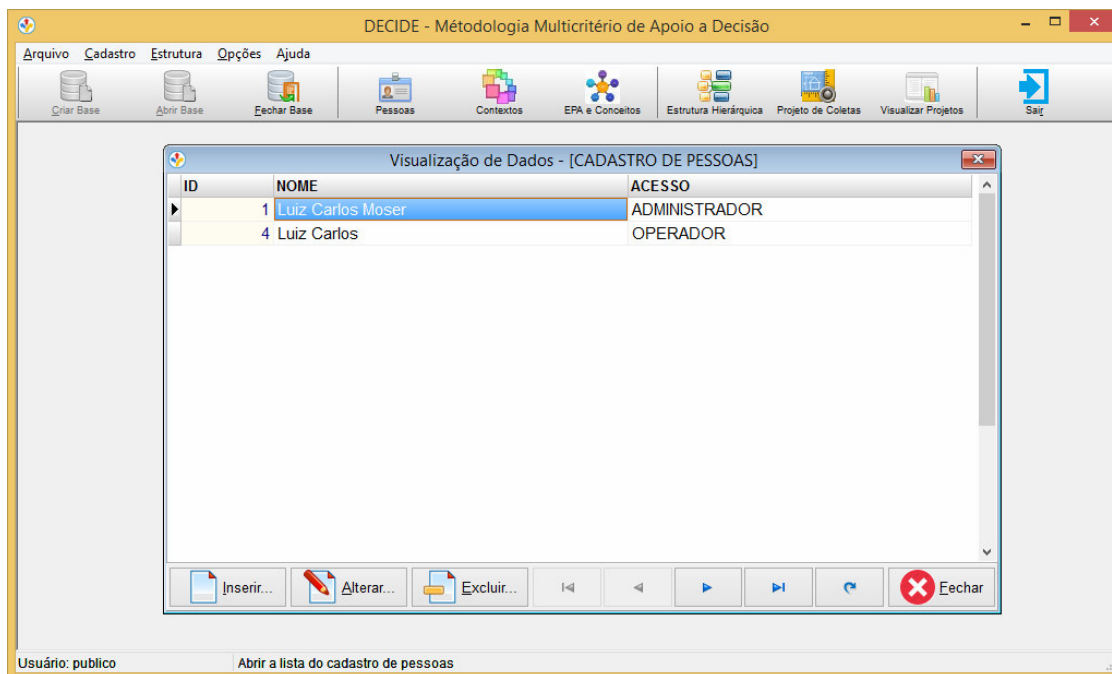


Figura 13 – Tela padrão de navegação

As ações do navegador de dados são atendidas pela classe de cadastro anteriormente passada como parâmetro para o construtor do navegador de dados e que em determinados casos pode ser instanciada gerando assim um objeto da classe de cadastro. Em programação orientada a objetos, chama-se instância de uma classe, um objeto cujo comportamento e estado são definidos pela classe e a instanciação é um processo por meio do qual se realiza a cópia de um objeto (classe) existente. Uma classe, que tem a função de determinar um tipo de dado, deve ser instanciada para que possa ser utilizada. Em outros termos, pode-se dizer que o navegador de dados recebe a classe de cadastro respectiva, armazena-os em uma variável na memória e somente instancia (cria o objeto) quando uma ação for solicitada pelo navegador de dados. Deste modo, o objeto criado a partir da classe de cadastro, faz a inicialização dos dados e prepara a tela de cadastro para a interação com o usuário.

Após a edição, os dados precisam ser gravados na base de dados por meio do botão “Salvar” que contém métodos abstratos implementados pela na classe estendida do objeto recém criado. A classe estendida faz a validação dos dados e só então permite ao método abstrato, que foi implementado, gravar na tabela da base de dados. Assim que os dados forem armazenados na base de dados, a classe de cadastro destrói o objeto de cadastro e retorna ao navegador informando que os dados foram alterados, ou inseridos para que seja atualizada a lista de registros. O botão cancelar simplesmente libera o objeto da classe de cadastro

retornando ao navegador com a informação de que nada foi feito. A Figura 14 mostra a tela do cadastro de pessoas gerenciada pela classe de cadastro padrão.

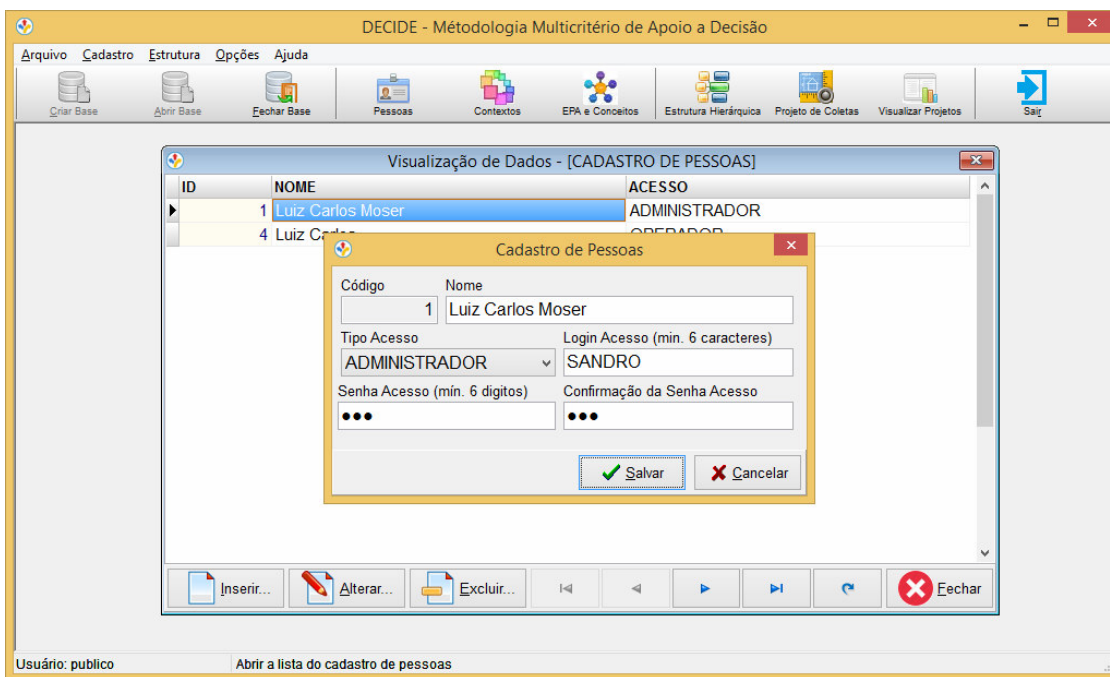


Figura 14 – Tela padrão de cadastros

A tela de cadastro de contexto é utilizada para registrar o contexto do problema, seus atores, as pessoas envolvidas no processo decisório. O sumário apresenta a importância, objetivo, metodologia e o resultado esperado. A tela de cadastro de contexto é apresentada na Figura 15.

Cadastro de Contexto

Contexto Atores Sumário Resumo

Apresentação do Problema
Distribuição de vagas de professores é uma atividade complexa para o gestor, pois envolve conflito de interesse entre os departamentos.

Importância
Desenvolver um modelo de avaliação de desempenho para apoiar o gestor nesta atividade é fundamental para dar transparência no processo de distribuição de vagas.

Objetivo
Construir um modelo de avaliação de desempenho que apóie o decisor no processo de distribuição de vagas.

Metodologia
Para construção do modelo será usado a metodologia MCDA-C.

Resultado
Espera-se ao final do trabalho um modelo de avaliação de desempenho que apóie o decisor no processo de distribuição de vagas de professores e que o processo torne-se transparente e maximize os objetivos da instituição.

✓ Salvar ✗ Cancelar

Figura 15 – Tela de cadastro de contexto

A tela de cadastro de EPA e Conceitos agrupa os conceitos por assunto abordado. Ainda, nesta tela, pode ser informada a data de cadastro para temporizar quando foi criado o agrupamento. Os agrupamentos servem para organizar os conceitos e não serão utilizados nas etapas seguintes. Os conceitos registrados nesta tela são utilizados pelo editor da estrutura hierárquica para compor a árvore do modelo decisório. A tela de EPA e Conceitos é apresentada na Figura 16.

Cadastro de EPA e Conceitos

Código: 8 Nome: Artigos publicados em periódicos Data de Cadastro: 26/11/2015

Relação de Conceitos (use TAB ou Ctrl+ENTER para adicionar na lista)

Garantir publicações em periódicos A1 e A2 ... deixar de publicar em periódicos de alto impacto

Adicionar

Remover

ID	NOME
10	Garantir publicações em periódicos A1 e A2 ... deixar de publicar em periódicos de alto impacto
11	Garantir publicações em periódicos B1 e B2 ... deixar de publicar em periódicos de alto impacto
12	Garantir publicações em periódicos B3 a B5 ... deixar de inserir os alunos na pesquisa
13	Garantir publicações em periódicos C ... deixar de inserir o aluno na pesquisa

Salvar Cancelar

Figura 16 – Tela de cadastro de EPAs e conceitos

Algumas telas apresentam estruturas diferentes das estruturas mencionadas nas telas de cadastros devido às particularidades técnicas, como o editor do modelo da estrutura hierárquica. O editor da estrutura hierárquica é desenvolvido com base em “Canvas” (componente para desenhar formas geométricas como círculos, linha e retângulo em uma janela), capturando as solicitações do sistema operacional, como mouse e teclado, e repassando aos elementos do editor afim de que o elemento interessado possa resolver o pedido. Ao receber a solução devolvida pelo elemento interessado, o editor devolve ao sistema operacional a resposta ao seu pedido.

O editor da estrutura hierárquica concentra a maior parte das regras de negócio e as mais complexas, incluindo cálculos de geometria, proporcionalidade, códigos para ordenação de dados, códigos para identificação espacial dos elementos na estrutura hierárquica, cálculos para distribuição em 2D (duas dimensões) dos elementos que se sobrepõem como por exemplo as linhas dos gráficos, valores das escalas ou valores dos indicadores. Nesse caso, quando há a sobreposição de elementos, é necessário que o algoritmo de ajuste calcule a posição disponível e faça o deslocamento apropriados para essa nova posição.

O editor da estrutura hierárquica cria o elemento principal (*root*) no primeiro nível a partir do rótulo cadastrado no contexto permitindo a inclusão de indicadores de segundo nível. Esses indicadores de segundo nível permitem relacionar os conceitos cadastrados na tela de “EPA/Conceitos” em grupos de interesse. Na sequência podem ser incluídos indicadores de terceiro nível os quais permitem relacionar os conceitos da lista relacionada no indicador de segundo nível. A Figura 17 apresenta o editor da estrutura hierárquica.

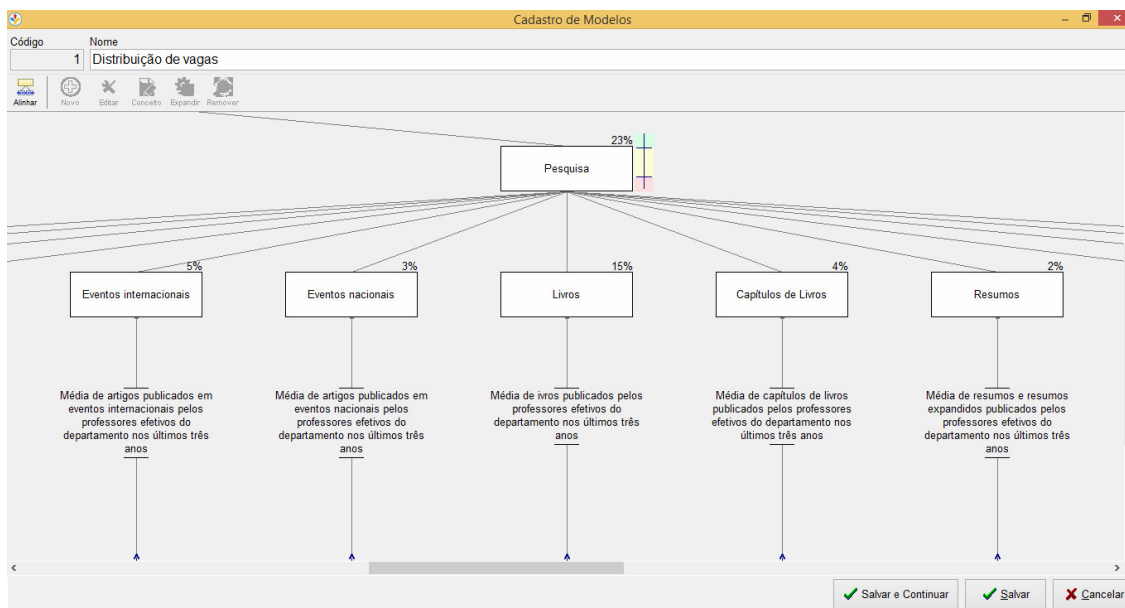


Figura 17 – Editor da estrutura hierárquica

Os indicadores e descritor da estrutura hierárquica são editados em editores auxiliares. As chamadas dos editores podem ser feitas pela barra flutuante de acesso rápido ou pela barra de botões no topo do editor.

O editor do indicador é responsável por permitir editar o texto do indicador e a taxa para cálculo do valor utilizado pelo indicador ascendente e é apresentado na Figura 18.

Figura 18 – Editor auxiliar do indicador da estrutura hierárquica

O editor auxiliar de conceitos do indicador é responsável por relacionar os conceitos do cadastro de EPAs e Conceitos ao indicador respectivo. O indicador de segundo nível apresenta os conceitos disponíveis, que não foram relacionados em outros indicadores, à esquerda e à direita os conceitos relacionados ao próprio indicador. Esses conceitos listados à direita são usados pelos indicadores de terceiro nível (indicadores filhos). No indicador de terceiro nível são apresentados, à esquerda, apenas os conceitos relacionados ao indicador ascendente e que ainda não foram usados por outros indicadores do mesmo nível, e à direita é apresentada a lista de conceitos relacionados ao indicador em questão. A Figura 19 apresenta o editor auxiliar dos conceitos do indicador.

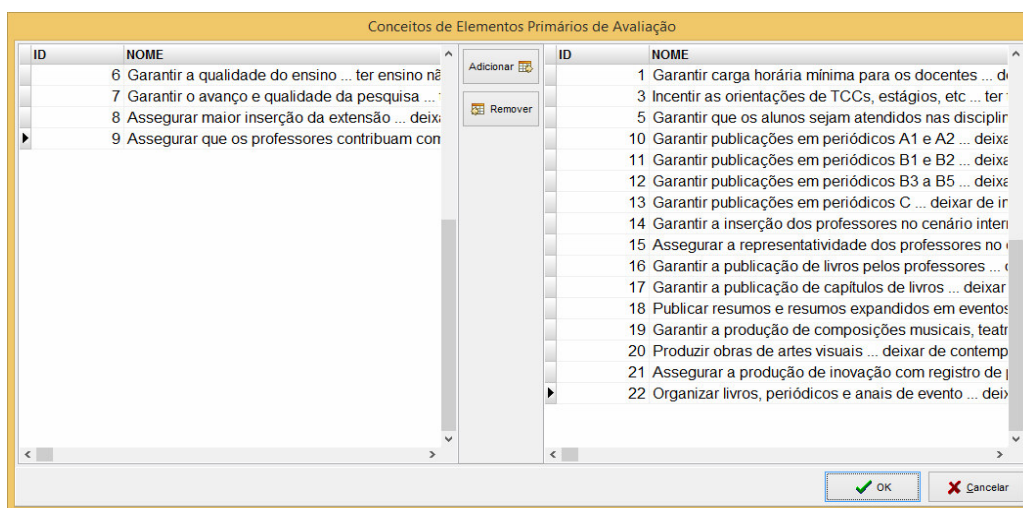


Figura 19 – Editor auxiliar dos conceitos do indicador da estrutura hierárquica

O editor auxiliar dos descritores é responsável pela composição da escala com os valores ordinais e cardinais. A Figura 20 apresenta o editor auxiliar de escalas dos descritores.

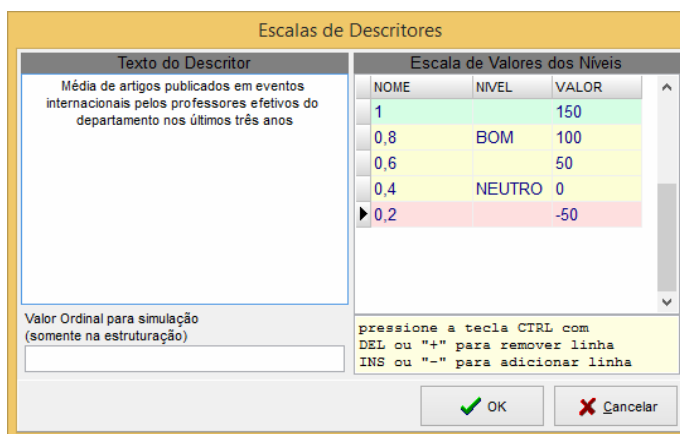


Figura 20 – Editor auxiliar das escalas dos descritores

O editor auxiliar da estrutura hierárquica é responsável por estender a estrutura hierárquica principal possibilitando a expansão dos conceitos em indicadores interligados e dispendo-os em qualquer posição livremente, diferentemente do editor da estrutura hierárquica principal. Nesse editor é permitido apenas editar o texto, a taxa de peso do indicador e incluir ou excluir novos indicadores.

Após a edição ser confirmada, cada indicador interligado ao conceito vai gerar um descritor permitindo assim configurar a escala para valoração ordinal e cardinal. A valoração ordinal pode ser quantitativa ou qualitativa enquanto que a valoração cardinal permite apenas valores quantitativos inteiros ou racionais. Na Figura 21 é apresentada a tela do editor auxiliar da estrutura hierárquica.

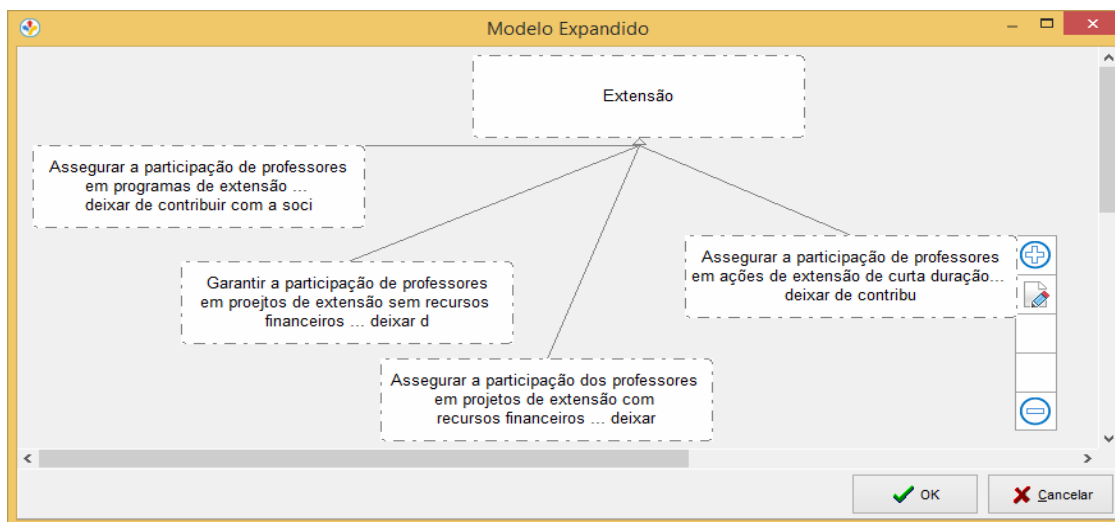


Figura 21 – Editor da estrutura hierárquica expandida

Após concluir a construção da estrutura hierárquica, o próximo passo é a definição de um projeto com a identificação do responsável e o modelo da estrutura hierárquica permitindo assim a valoração dos descritores. A Figura 22 apresenta o cadastro do projeto.

Cadastro de Projetos

Código: 5 Nome do Projeto: Informática Data do Projeto: 26/11/2015

Responsável pelo Projeto: Luiz Carlos Moser Modelo Padrão do Projeto (Estrutura Hierárquica): Distribuição de vagas

Dados dos Descritores

Código	Descritor	INTERVALO	VALOR
85	Média de orientações de TCC e Estágio supervisionando realizados pelos pro	2 - 6	5,32
86	Média de alunos atendidos pelos professores efetivos do departamento dos i	60 - 100	65,81
87	Média de carga horária (h/a) semanal dos professores efetivos do departame	8 - 12	12,06
88	Média de artigos A1 e A2 publicados pelos professores efetivos do departarr	0,2 - 1	0,08
89	Média de artigos B1 e B2 publicados pelos professores efetivos do departarr	0,2 - 1	0,45
90	Média de artigos B3 e B5 publicados pelos professores efetivos do departarr	0,2 - 1	0,56
91	Média de artigos C publicados pelos professores efetivos do departamento n	0,4 - 2	1,69
92	Média de artigos publicados em eventos internacionais pelos professores efe	0,2 - 1	1,05
93	Média de artigos publicados em eventos nacionais pelos professores efetivos	1 - 3	0,34
94	Média de ivros publicados pelos professores efetivos do departamento nos ú	0,2 - 1	0,08
95	Média de capítulos de livros publicados pelos professores efetivos do departa	0,2 - 1	0,11
96	Média de resumos e resumos expandidos publicados pelos professores efeti	0,5 - 2,5	1,46
97	Média de composições musicais, peça teatral e filme realizados pelos profes	0,2 - 1	0

Figura 22 – Cadastro do projeto

Os valores lançados no cadastro de projeto podem ser contrapostos com a estrutura hierárquica produzindo o mapa para a tomada de decisão na tela de Visualização do Mapa. Para isso é necessário selecionar um modelo de estrutura hierárquica para apresentação de todos os projetos que usam o mesmo modelo de estrutura hierárquica e na sequência deve ser selecionado apenas os projetos que serão visualizados no mapa decisório. A Figura 23 apresenta a tela de seleção de projetos.

Seleção do Projeto para Visualização no Mapa

Selecione o Modelo de Estrutura Hierárquica para filtrar os projetos relacionados

Distribuição de vagas

Visualizar?	NOME DO PROJETO	DATA PROJETO
<input type="checkbox"/>	NAO Agrimensura	26/11/2015 13:36:34
<input type="checkbox"/>	NAO Eng. Construção Civil	26/11/2015 13:36:34
<input type="checkbox"/>	NAO Eng. Mecânica	26/11/2015 13:36:34
<input type="checkbox"/>	SIM Física	26/11/2015 13:36:34
<input type="checkbox"/>	SIM Informática	26/11/2015 13:36:34
<input checked="" type="checkbox"/>	NAO Matemática	26/11/2015 13:36:34

Figura 23 – Tela de seleção de projetos

A visualização do mapa decisório é responsável pela contraposição dos valores da estrutura hierárquica com os valores informados no projeto de coletas. A Figura 24 apresenta a tela de visualização do mapa decisório.

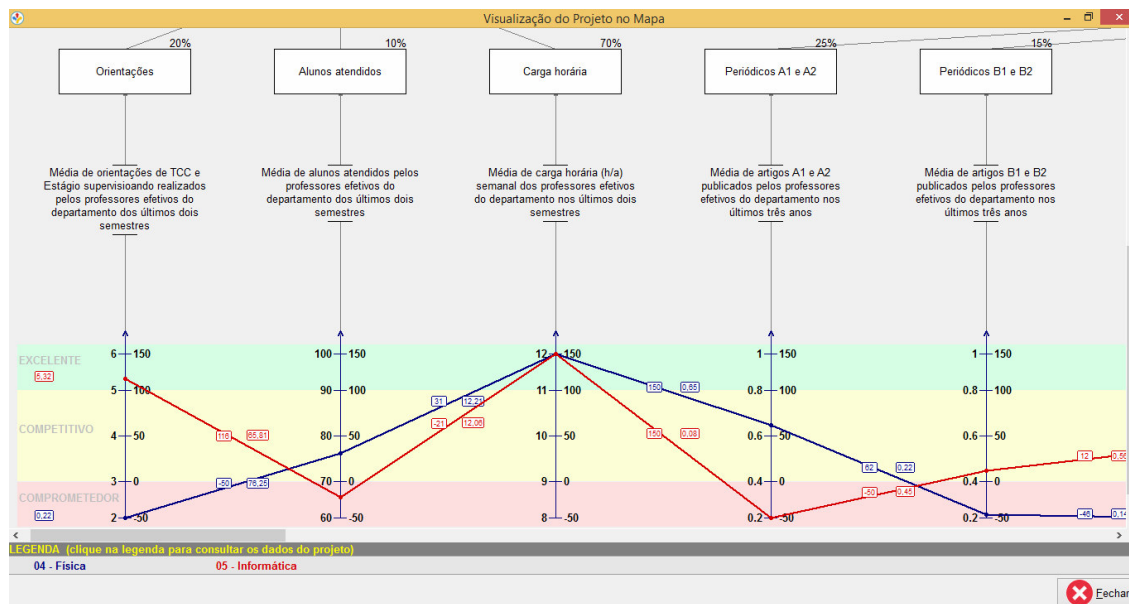


Figura 24 – Visualização do mapa decisório

4.4 IMPLEMENTAÇÃO DO SISTEMA

Como apresentado na Seção 4.3 o layout do sistema é composto por um setor principal, a tela do inicial do sistema.

Para facilitar o desenvolvimento foram criados alguns padrões que incluem uma classe de cadastro padrão, uso de ações (*actions*) para controlar o acesso aos menus e barra de botões da tela principal e alguns componentes para uso do editor da estrutura hierárquica. Além disso, botões maiores para facilitar o uso do mouse, fonte de letras em tamanho maior para facilitar a leitura, disposição padronizada dos elementos para todas as telas.

Na classe principal do sistema, encontram-se as ações (*actions*) para controlar o acesso as opções do sistema. O método `ActionListUpdate` é responsável por habilitar ou desabilitar as opções do menu principal e da barra de botões que estão diretamente ligadas com as ações. A Listagem 1 apresenta o código responsável por controlar o acesso às opções do sistema.

```

1  procedure TfmMain.ActionListUpdate(Sender: TObject);
2  var
3      bActive: Boolean;
4      ac: TAction;
5  begin
6      bActive := dmDados.DataActive;
7
8      // Ativar ou desativar cada item do menu e toolbar
9      ac := TAction(Sender);
10
11     if ac = nil then // aqui comparado para evitar que um Action seja chamado manualmente.
12         Exit
13
14     // Autenticação
15     else if ac = acArquivoAutenticar then
16     begin
17         ac.Enabled := bActive and Sistema.UserAdmin;// or not Sistema.Config.UserLoginRequired);
18         ac.Checked := bActive and Sistema.Config.UserLoginRequired;
19     end
20
21     // Arquivo
22     else if ac = acArquivoNovo then
23         ac.Enabled := not bActive
24     else if ac = acArquivoAbrir then
25         ac.Enabled := not bActive
26     else if ac = acArquivoFechar then
27         ac.Enabled := bActive
28
29     // Cadastro
30     else if ac = acCadastroPessoa then
31         ac.Enabled := bActive and Sistema.UserAdmin
32     else if ac = acCadastroContexto then
33         ac.Enabled := bActive and Sistema.UserGerente

```

Listagem 1 – Método ActionListUpdate

Como apresentado na Listagem 1 cada ação (*action*) envia o método ActionListUpdate para controlar alguns atributos. Cada botão ou menu da tela principal está diretamente ligada a um desse *action* e, portanto, herda o atributo da ação. Dessa forma, só é preciso atribuir um valor para a ação e o menu ou botão também utiliza o mesmo valor em seus atributos.

Outro método importante é o ActionListExecute que é responsável por executar as ações de clique no menu principal ou no botão da barra de topo ao qual estão associados. Na Listagem 2 é apresentado o código do método.

```

1  procedure TfmMain.ActionListExecute(Sender: TObject);
2  var
3      ac: TAction;
4  begin
5      // Ação do Action disparado pelo menu ou toolbar
6      ac := TAction(Sender);
7
8      if ac = nil then // aqui comparado para evitar que um Action seja chamado manualmente.
9          Exit
10
11     // Arquivo
12     else if ac = acArquivoNovo then
13         dmDados.DataNewDlgDB(True)
14     else if ac = acArquivoAbrir then
15         dmDados.DataOpenDlgDB()
16     else if ac = acArquivoFechar then
17         begin
18             if PermiteFecharBaseDados(false) then
19                 dmDados.DataCloseDB();
20         end
21
22     // Cadastro
23     else if ac = acCadastroPessoa then
24         ShowGrid(TfmCadPessoa)
25     else if ac = acCadastroContexto then
26         ShowForm(TfmCadContexto)
27     else if ac = acCadastroEPA then
28         ShowGrid(TfmCadConceitoEPA)
29
30     // Futuramente

```

Listagem 2 – Método ActionListExecute

O cadastro padrão é composto por uma classe herdada por quase todos os cadastros do sistema e contém código de uso comum devidamente preparados para otimizar o reuso de código. Essa classe de cadastro trabalha em conjunto com a *grid* de navegação por meio de inversão de controle. A *grid* de navegação se comunica com a classe de cadastro que por sua vez responde às solicitações ou transfere essa responsabilidade para as classes de cadastro que a herdaram. Desta forma, as ações são padronizadas para todos os cadastros. O código da declaração da classe padrão de cadastro é apresentado na Listagem 3.


```

1  TfmCrudPadrao = class(TForm)
2      pnBotoes: TPanel;
3      btnCancelar: TBitBtn;
4      btnOK: TBitBtn;
5      procedure btnOKClick(Sender: TObject);
6      procedure btnCancelarClick(Sender: TObject);
7      procedure FormCreate(Sender: TObject);
8      procedure FormDestroy(Sender: TObject);
9      procedure FormResize(Sender: TObject);
10     private
11     //      function ShowModal: Integer;
12     { Private declarations }
13     protected
14     // funções de estruturação
15     KeyID      : Integer; // 0: Novo; >0: Editar
16     KeyData    : TFDQuery;
17     KeySource  : TDataSource;
18     public
19     function LoadKeyID(const table: AnsiString; const id: Integer=KEYDATA_NEW): Boolean; overload; virtual;
20     function LoadKeyID(const table: AnsiString; const DataSet: TFDQuery): Boolean; overload; virtual;
21     function SaveKeyID(const add: Boolean): Boolean; virtual;
22
23     // >> CRUD - Métodos - DataSet sempre ReadOnly (VIEW) <<
24     // métodos da classe (não precisa do objeto instanciado para executar)
25     class procedure Listar(const DataSet: TFDQuery); virtual; abstract;
26     class function Preparar(const DataSet: TFDQuery; var Titulo: String): Boolean; virtual; abstract;
27     class function Excluir(const DataSet: TFDQuery): Boolean; virtual; abstract;
28
29     // métodos CRUD (cria o objeto, faz a edição de dados)
30     function Validar(const Add: Boolean): Boolean; virtual; abstract;
31     function Gravar(const Add: Boolean): Boolean; virtual; abstract;
32     function Inserir(const DataSet: TFDQuery): Boolean; virtual; abstract;
33     function Editar(const DataSet: TFDQuery): Boolean; virtual; abstract;
34
35     constructor Create(AOwner: TComponent); override;
36     function ShowModal: Integer; override;
37
38     end;

```

Listagem 3 – Código da declaração da classe padrão de cadastro

Na Listagem 3 estão os métodos abstratos para serem implementados pelas classes que à herdam e devem providenciar o processamento das ações solicitadas pela *grid* de navegação de dados.

Os dados podem ser carregados da tabela da base de dados usando o método LoadKeyID para edição e posteriormente gravados nas mesmas tabelas com o uso do método SaveKeyID. Esses métodos fazem o trabalho repetitivo das classes de cadastro e são apresentados na Listagem 4.

```

1  function TfmCrudPadrao.LoadKeyID(const table: AnsiString; const id: Integer=KEYDATA_NEW): Boolean;
2  begin
3      KeyData.Close;
4      UpdateDataLinks(self, KeySource);
5      if id = KEYDATA_FULL then // todos os registros (sem filtro)
6          KeyData.Open('SELECT * FROM ' +table )
7      else
8          KeyData.Open('SELECT * FROM ' +table+ ' WHERE ID_'+table+'='+ IntToStr(id));
9
10     // Mostrar o Código da chave primaria da tabela (automaticamente)
11     result := PrepareViewFields(KeyData,table);
12     // UpdateDataLinks(self, KeySource);
13     SelectFirst;
14 end;
15
16 function TfmCrudPadrao.SaveKeyID(const add: Boolean): Boolean;
17 begin
18     if (KeyData.State in dsEditModes) then
19         KeyData.Post;
20
21     result := not KeyData.IsEmpty;
22 end;

```

Listagem 4 – Métodos para ler e gravar dados nas tabelas de cadastro

As classes de cadastro que herdam a classe padrão podem sobrescrever ambos os métodos LoadKeyID e SaveKeyID e implementar modificações se houver necessidade.

A *grid* de navegação de dados é preparada para interagir com a classe de dados preparando a tela de navegação de dados. Na Listagem 5 está o código que implementa essa funcionalidade.

```

1  constructor TfmDadosGrid.Create(const ViewClassInput: TfmCrudPadraoClass);
2  var
3      s : string;
4      FM : TCustomForm;
5      pt : TPoint;
6  begin
7      inherited Create(Application.MainForm);
8      FViewClass := ViewClassInput;
9
10     // >> Preparar os dados da consulta
11     if FViewClass.Preparar(qryConsulta, s) then
12     begin
13         Caption := 'Visualização de Dados - ['+s+']';
14
15         // --> cada tela abre seu próprio clientdataset e seu sqlquery
16         AdjusVisibleFields;
17         AdjustColumnWidths;
18         if grdDados.Columns.Count > 0 then
19             grdDados.SelectedIndex := 1;
20     end;
21 end;

```

Listagem 5 – Implementação da funcionalidade da grid de navegação de dados

Com os dados carregados no *grid* de navegação, o usuário pode editar as informações selecionando um registro e clicando no botão de Alterar. Na Listagem 6 é apresentado o código do método Editar.

```

1  procedure TfmDadosGrid.btnEditarClick(Sender: TObject);
2  var
3      View: TfmCrudPadrao;
4  begin
5      if Assigned(FViewClass) then
6      begin
7          if dsConsulta.DataSet.IsEmpty then
8              MsgErro('Não há nenhum item selecionado para editar.')}
9          else
10         begin
11             View := FViewClass.Create(self);
12             try
13                 try
14                     if View.Editar(qryConsulta) then
15                         dsConsulta.DataSet.Refresh // recarregar
16                     else if View.ModalResult = mrOk then // botão OK, mas não gravou (Cancelar não gera erro)
17                         MsgErro('Houve erro ao alterar o registro.')}
18                 except
19                     if qryConsulta.State in dsEditModes then // para os casos em que a grid permite editar
20                         qryConsulta.CancelUpdates;
21                         raise;
22                     end;
23                 finally
24                     View.Release;
25                 end;
26             end;
27         end;
28     end;

```

Listagem 6 – Código do método editar

No *grid* de navegação o usuário pode inserir novos registro, para isso deve clicar no botão Inserir. Na Listagem 7 é apresentado o código do método Inserir.

```

1  procedure TfmDadosGrid.btnInserirClick(Sender: TObject);
2  var
3      View: TfmCrudPadrao;
4  begin
5      if Assigned(FViewClass) then
6      begin
7          View := FViewClass.Create(self);
8          try
9              try
10                 if View.Inserir(qryConsulta) then
11                     dsConsulta.DataSet.Refresh // recarregar
12                 else if View.ModalResult = mrOk then // botão OK, mas não gravou
13                     MsgErro('Houve erro ao inserir o registro.')}
14             except
15                 if qryConsulta.State in dsEditModes then // para os casos em que a grid permite editar
16                     qryConsulta.CancelUpdates;
17                     raise;
18                 end;
19             finally
20                 View.Release;
21             end;
22         end;
23     end;
24

```

Listagem 7 – Código do método inserir

No *grid* de navegação o usuário pode excluir um registro selecionado clicando no botão Excluir. Na Listagem 8 é apresentado o código do método excluir.

```

1  procedure TfmDadosGrid.btnRemoverClick(Sender: TObject);
2  begin
3      if Assigned(FViewClass) then
4      begin
5          if dsConsulta.DataSet.IsEmpty then
6              MsgErro('Não há nenhum item selecionado para excluir.')}
7          else
8              begin
9
10             if MsgOK('Excluir definitivamente o item selecionado?') then
11             begin
12                 if FViewClass.Excluir(qryConsulta) then
13                     dsConsulta.DataSet.Refresh
14                 else
15                     MsgErro('Houve erro ao excluir o registro.')}
16             end;
17         end;
18     end;
19 end;
20 end;

```

Listagem 8 – Código do método excluir

As classes de cadastros implementam os métodos abstratos da classe padrão de cadastro que respondem pela ação solicitada da *grid* de navegação de dados. O método Preparar é chamado pela *grid* de navegação para carregar os dados que são listados. Na Listagem 9 é apresentado o código do método Preparar implementado pela da classe Pessoa.

```

1  class function TfmCadPessoa.Preparar(const DataSet: TFDQuery; var Titulo: String): Boolean;
2  begin
3      Titulo := 'CADASTRO DE PESSOAS';
4
5      DataSet.SQL.Text := 'SELECT * FROM PESSOA';
6      DataSet.Open;
7      DataSet.FieldByName('LOGIN_PESSOA').Visible := false;
8      DataSet.FieldByName('SENHA_PESSOA').Visible := false;
9
10     result := PrepareViewFields(DataSet, 'PESSOA');
11 end;

```

Listagem 9 – Código do método Preparar implementado pela classe Pessoa

O método Validar é chamado sempre que for solicitado para gravar os dados na tabela da base de dados e antes da chamada do método Gravar. Esse método é responsável por fazer a validação dos dados sendo implementado por cada classe de cadastro que herdar da classe de cadastro padrão. Quando o método validar retornar um valor falso como resposta, o sistema ignora a chamada ao método Gravar retornando o controle para a tela de cadastro sem nada fazer. Na Listagem 10 é apresentado o código do método Validar.

```

1  function TfmCadPessoa.Validar(const Add: Boolean): Boolean;
2  begin
3      result := False;
4
5      if edNome.Text = '' then
6          MsgErro('Informe o nome da Pessoa')
7      else if edTipoAcesso.ItemIndex < 0 then
8          MsgErro('Informe o tipo de acesso')
9      else if edTipoAcesso.ItemIndex > 0 then // sendo 0 é acesso não informado (quando
10         begin
11             if edLogin.Text= '' then
12                 MsgErro('Informe um login para acesso ao sistema')
13             else if Length(edLogin.Text) < 3 then
14                 MsgErro('Informe pelo menos 3 caracteres para o login')
15             else if edSenha.Text = '' then
16                 MsgErro('Informe uma senha para acesso ao sistema')
17             else if Length(edSenha.Text) < 3 then
18                 MsgErro('Informe pelo menos 3 caracteres para a senha')
19             else if edSenha.Text <> edConfirma.Text then
20                 MsgErro('A senha informada não confere com a senha de confirmação')
21             else
22                 result := True;
23         end;
24     end;

```

Listagem 10 – Código do método Validar

O método Inserir é chamado sempre que for solicitado pela *grid* de navegação para criar um novo registro na tabela da base de dados. Esse método é responsável por carregar os campos vazios e inicializar os componentes e abrir a respectiva tela da classe de cadastro. Na Listagem 11 é apresentado o código do método Inserir.

```

1  function TfmCadPessoa.Inserir(const DataSet: TFDQuery): Boolean;
2  begin
3      LoadKeyID('PESSOA');
4      KeyData.FieldByName('SENHA_PESSOA').OnGetText := DoDecryptSenha;
5      KeyData.FieldByName('SENHA_PESSOA').OnSetText := DoEncryptSenha;
6      KeyData.First; // para recarregar "senha" e executar onGetText
7
8      edConfirma.Clear;
9      edTipoAcesso.ItemIndex := 0;
10     UpdateEnableds;
11
12     result := (ShowModal = mrOk);
13 end;

```

Listagem 11 – Código do método Inserir

O método Editar é chamado sempre que for solicitado pela *grid* de navegação para editar os dados na tabela da base de dados. Esse método é responsável por carregar os campos com seus respectivos valores nos componentes e abrir a respectiva tela da classe de cadastro. Na Listagem 12 é apresentado o código do método Editar.

```

1  function TfmCadPessoa.Editar(const DataSet: TFDQuery): Boolean;
2  begin
3      LoadKeyID('PESSOA', DataSet);
4      KeyData.FieldByName('SENHA_PESSOA').OnGetText := DoDecryptSenha;
5      KeyData.FieldByName('SENHA_PESSOA').OnSetText := DoEncryptSenha;
6      KeyData.First; // para recarregar "senha" e executar onGetText
7
8      edConfirma.Text := edSenha.Text;
9      UpdateEnableds;
10
11     result := (ShowModal = mrOk);
12 end;

```

Listagem 12 – Código do método Editar

O método Gravar é chamado sempre que a tela de cadastro solicitar salvar as alterações feitas nos campos de dados anteriormente carregados pelo método Inserir ou Alterar. Algumas funções executam códigos genéricos de uso comum. O método SaveKey recebe no parâmetro Add o valor True indicando se salvará os dados carregados pelo método Inserir ou recebe um valor False indicando que vai salvar os dados carregados pelo método Alterar. O método SaveKey faz as devidas verificações, encontra o campo na tabela e faz a gravação dos dados. Na Listagem 12 é apresentado o código do método Gravar.

```

1  function TfmCadPessoa.Gravar(const Add: Boolean): Boolean;
2  begin
3      result := SaveKeyID(Add);
4  end;

```

Listagem 13 – Código do método Gravar

As classes responsáveis pelo editor da estrutura hierárquica, pelos indicadores e pelos descritores contêm os cálculos e códigos de processamento para produzir o resultado final, a estrutura hierárquica. O código *canvas* para desenhar o indicador no editor da estrutura hierárquica é apresentado na Listagem 14.

```

9  itIndicador:
10 begin
11     // a diferença (margem) deve ser excluída de cada lado
12     n := (ITEM_COLSIZE - ITEM_WIDTH) div 2; // margens (para compatibilizar com tamanho dos decisores)
13     InflateRect(RBox, -n, 0);
14
15     // valor do peso no topo a direita do indicador
16     if Owner is TElementItem then // elemento root não tem valueRate
17     begin
18         PrintText( Canvas, FormatFloat('#0',FValueRate)+'%', RBox, DT_SINGLELINE or DT_RIGHT or DT_TOP );
19     end;
20     Inc( RBox.Top, ITEM_TEXTTOP ); // saltar o espaço de topo (valor do peso)
21
22     // limpar o fundo e pintar a area do elemento
23     Canvas.Brush.Color := clWhite;
24     Canvas.Pen.Color := clBlack;
25     Canvas.Pen.Style := psSolid;
26
27     Canvas.Rectangle(RBox);
28
29     // escalas mini mapas (somente se não for indicador contendo um decisor)
30     if (ComponentCount > 0) then
31     begin
32         if (ComponentCount > 1) or (TElementItem(Components[0]).ItemType <> itDescriptor) then
33         begin
34             // aqui o mini mapa vai ocupar metade do espaço a direita do indicador e o valor vai a sua direita
35             RTmp := Rect(RBox.Right + 2, RBox.Bottom - getPositionSize(false), RBox.Right + n - (n div 2), RBox.Bottom);
36             DrawMapScale( RTmp, nil );
37             DrawMapValue( RTmp, false );
38         end;
39     end;
40 end;

```

Listagem 14 – Código canvas do indicador

O código *canvas* para desenhar o descritor no editor da estrutura hierárquica é apresentado na Listagem 15.

```

52  itDescriptor:
53  begin
54      Canvas.Brush.Color := clBtnFace;
55      Canvas.FillRect(RBox);
56
57      // linha central vertical do elemento (transversal)
58      Canvas.Pen.Color := clGray;
59      Canvas.Pen.Style := psSolid;
60      Canvas.MoveTo(RBox.Left + (RBox.Width div 2), RBox.Top );
61      Canvas.LineTo(RBox.Left + (RBox.Width div 2), RBox.Bottom);
62
63      RTmp := Rect(RBox.Left, RBox.Bottom - getPositionSize(false), RBox.Right, RBox.Bottom);
64      // escalas detalhadas
65      DrawMapScale( RTmp, FScaleList );
66      // valores das escalas e linhas de ligação entre decisores (esquerda e direita)
67      DrawMapValue( RTmp, true );
68  end;

```

Listagem 15 – Código canvas do descritor

5 CONCLUSÃO

A MCDA-C empregada no desenvolvimento da ferramenta de apoio ao gestor adota uma abordagem construtivista, sendo assim, o modelo construído não busca identificar uma solução definitiva e absoluta, mas fazer com que o decisor tenha maior conhecimento sobre o seu problema, o que permitirá identificar os pontos de aperfeiçoamento. Outro aspecto importante a ser considerado, se for adotada a postura construtivista, é a geração de conhecimento evolutivo dinâmico e contínuo, não há um ponto final demarcado previamente no processo.

Ainda há necessidade de considerar que a limitação do processo é condicionada à representação do problema pelo decisor proporcionando resultados apenas na abordagem representada no modelo construído.

Diante da necessidade de gestões mais eficientes, é essencial que as instituições de ensino busquem meios mais eficazes de aplicação de recursos. A MCDA-C, como metodologia de apoio à decisão, pode auxiliar no processo de estruturação e avaliação do desempenho institucional. E o uso dessa metodologia é facilitado pelo software desenvolvido como resultado deste trabalho.

Durante o desenvolvimento do software, várias foram as dificuldades encontradas para adequar-se aos requisitos do cliente principalmente pelo uso de *canvas* e o processamento gráfico com as estruturas geométricas. Nem sempre é possível prever como a ferramenta de software vai se comportar em um cenário WYSIWYG (acrônimo da expressão em inglês What You See Is What You Get, cuja tradução remete a algo como "O que você vê é o que você obtém").

O resultado final foi uma ferramenta robusta, que agilizará a construção de projetos baseados na metodologia MCDA-C permitindo fazer alterações tanto na estrutura quando nos valores inseridos no projeto e obtendo a visualização imediata do mapa decisório. Além da agilidade no processo, a ferramenta proporciona a comparação entre vários projetos em um único mapa decisório, tornando-se um referencial e desta forma fornecendo uma visão mais ampla da questão abordada pelo decisor no processo de avaliação das recomendações.

Como trabalhos futuros destacam-se relatórios que possam ser úteis ao gestor e ajustes no sistema para complementar a funcionalidade de negócio implementada.

REFERÊNCIAS

BANA E COSTA, Carlos A. Três convicções fundamentais na prática do apoio à decisão. **Pesquisa operacional**, v. 13, p. 1-12, 1993.

BANA E COSTA, Carlos A., SILVA, Fernando N. Concepção de uma “boa” alternativa de ligação ferroviária ao porto de Lisboa: uma aplicação da metodologia multicritério de apoio à decisão e à negociação. **Investigação Operacional**, v. 14, p. 115-131, 1994.

BLAHA, Michael; JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2ª ed. Rio de Janeiro: Elsevier, 2006.

ENSSLIN, Leonardo; DUTRA, Ademar; ENSSLIN, Sandra R. MCDA: a constructivist approach to the management of human resources at a governmental agency. **International Transactions in Operational Research**, 2000, v. 7, p. 79-100.

ENSSLIN, Leonardo; GIFFHORN, Edilson; ENSSLIN, Sandra Rolim; PETRI, Sérgio Murilo; VIANNA, William Barbosa. Avaliação do desempenho de empresas terceirizadas com o uso da metodologia multicritério de apoio à decisão – construtivista. **Pesquisa Operacional**, v. 30, n. 1, p.125-152, Janeiro a Abril de 2010, p. 125-152.

ENSSLIN, Leonardo; MONTIBELLER NETO, Gilberto; NORONHA, Sandro M. **Apoio à decisão: metodologias para estruturação de problemas e avaliação multicritério de alternativas**. Florianópolis: Insular, 2001.

ENSSLIN, Sandra R. **Incorporação da perspectiva sistêmico-sinérgica na metodologia MCDA construtivista: uma ilustração de implementação**. Tese (Doutorado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

FRAIDENRAICH, Verônica. 5 pontos importantes sobre PPP e gestão financeira. 2010. **Gestão escolar**, ed. 7, Abril/Maio 2010. Disponível em: <<http://gestaoescolar.abril.com.br/administracao/ppp-gestao-financeira-caminham-juntos-551882.shtml>>. Acesso em: 11 nov. 2014.

KEENEY, Ralph L. **Value focused-thinking: A Path to Creative Decision-making**. Cambridge: Harvard Univ. Press, 1992.

LANDRY, Maurice. A note on the concept of problem. **Organization Studies**, v.16, p. 315-343, 1995.

MACBETH. **A multiple criteria decision support system**. Disponível em: <<http://www.m-macbeth.com/pt/m-home.html>>. Acesso em: 26 de nov. 2015.

PRESSMAN, Roger. **Engenharia de software**, 5ª ed. 2002. Rio de Janeiro: McGrawHill.

ROY, Bernard. **Multicriteria methodology for decision aiding**. Dordrecht: Kluwer Academic Publishers, 1996.

SCHNORRENBARGER, Darci. **Identificando, avaliando e gerenciando os recursos organizacionais tangíveis e intangíveis por meio de uma metodologia construtivista (MCDA): uma ilustração na área econômica-financeira**. 2005. Tese (Doutorado em Engenharia de Produção) – Curso de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2005.

SKINNER, Wickham. The productivity paradox. **Management Review**, n. 75, p. 41-45, 1986.

TEZZA, Rafael; ZAMCOPÉ, Fábio Cristiano; ENSSLIN, Leonardo. **A metodologia multicritério de apoio à decisão construtivista para a identificação e avaliação de habilidades para o setor de estampa textile**. GEPROS: Gestão da Produção, Operações e Sistemas – Ano 5, nº 1, Jan-Mar/2010, p. 125-142.

ZANELLA, Ítalo J. **As problemáticas técnicas no apoio à decisão**, Dissertação de Mestrado em Engenharia de Produção - Universidade Federal de Santa Catarina, Brasil, 1996.