



**DISSERTAÇÃO DE MESTRADO EM SISTEMAS
MECATRÔNICOS**

**Proposta do emprego de aprendizado de
máquinas como ferramenta de apoio a fase de
projeto conceitual**

Por,

João Marcelo Teixeira e Silva Torres

Brasília, 30 de Maio de 2016

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA MECANICA
UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Departamento de Engenharia Mecânica

DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS

Proposta do emprego de aprendizado de máquinas como ferramenta de apoio a fase de projeto conceitual

POR,

João Marcelo Teixeira e Silva Torres

Relatório submetido ao departamento de Engenharia Mecânica da Faculdade de Tecnologia da Universidade de Brasília como parte dos requisitos para obtenção do grau de Mestre em Sistemas Mecatrônicos.

Banca Examinadora

Profa. Andrea Cristina dos Santos, UnB/ EPR (Orientador)

Profa. Claudia de Oliveira Melo, UnB/ EPR (Examinadora Externa)

Prof. Rudi Henri Van Els, UnB/ FGA (Examidor interno)

Brasília, 15 de Dezembro de 2016

Dedicatória(s)

Dedico este trabalho aos meus pais, que me ensinaram a buscar o conhecimento e o trabalho como forma de aprimoramento pessoal. Também dedico as pessoas que gostaria que estivessem aqui neste momento, mas que hoje só podem estar presentes em nossas lembranças.

Vô Domingos, Salmo e Rafael.

João Marcelo Teixeira e Silva Torres

Agradecimentos

Agradeço novamente à meus pais, meus irmãos, irmã e minha noiva que sempre me apoiaram, encorajaram e se preocuparam comigo. A força que vocês me deram me impulsionou ao longo deste caminho.

Aos meus colegas e amigos que me acompanharam durante o trajeto, suando, rindo e acima de tudo, aprendendo.

Ao meu amigo Jones Yudi, que é um verdadeiro poço de ideias (boas e ruins).

Aos alunos da disciplina de PSP 6 do segundo semestre de 2016 da Universidade de Brasília que se dispuseram a ajudar neste projeto.

E por último, minha orientadora, a Professora Andrea Santos pela paciência, auxílio, encorajamento e lições durante todos estes anos, curtos em tempo mas longos em experiência.

Este trabalho também é de vocês. Espero que o apreciem.

João Marcelo Teixeira e Silva Torres

RESUMO

O projeto mecatrônico é complexo e envolve uma grande quantidade de conhecimento de várias áreas da Engenharia e Tecnologia da informação para alcançar os requisitos funcionais do sistema. Informações de projetos anteriores podem auxiliar no desenvolvimento de novos projetos, provendo agilidade e auxiliando no processo criativo de geração de princípios de solução. O objetivo deste trabalho é desenvolver um sistema automatizado para sugestão de soluções para a fase conceitual de projeto mecatrônico de forma a propor soluções melhores e que possam agilizar o processo decisório, utilizando conceitos de aprendizado de máquina para que o sistema pudesse evoluir com o uso continuado. Ao fim da pesquisa um protótipo deste sistema foi produzido e testado academicamente.

ABSTRACT

Mechatronic design is complex and involves a great amount of knowledge from several Engineering areas and Information Technology to achieve the System's functional requirements. Information from previous projects can assist in the development of new projects, providing agility and assisting in the creative process of generating new solution principles. The objective of this research is to develop an automated system capable of suggesting solutions for the conceptual phase of mechatronic design in order to propose better solutions that can speed up the decision process using machine learning concepts allowing the system to evolve with continued use. At the end of the research a prototype of this system was developed and tested academically.

SUMÁRIO

1 INTRODUÇÃO	12
1.1 Objetivo da pesquisa	17
1.1.1 Objetivos Específicos	17
1.2 Metodologia da pesquisa.....	17
1.3 Estrutura do trabalho	20
2 REVISÃO BIBLIOGRÁFICA.....	22
2.1 O Projeto na Engenharia Mecatrônica.....	22
2.2 A Fase de Projeto conceitual.....	23
2.3 Aprendizado de máquina.....	28
2.3.1 Conceitos e utilização	28
2.3.2 Raciocínio Baseados em Casos	32
2.3.3 Principais métodos e paradigmas para RBC.....	35
2.3.4 Motivação para o uso do RBC	39
2.4 <i>Computer Aided Innovation (CAI)</i>	41
3 Desenvolvimento.....	43
3.1 Solução proposta	43
3.2 Requerimentos do software.....	44
3.3 Dependências externas.....	45
3.4 Arquitetura de software	46
3.4.1 Modelo	49
3.4.1.1 Implementação	50
3.4.2 Controle.....	53
3.4.3 Visão	61
3.4.3.1 Implementação	76

4	Resultados	78
4.1	Metodologia de avaliação	78
4.1.1	Validação	79
4.1.1.1	Ambiente de testes	81
4.1.2	Verificação	82
4.1.2.1	Ambiente de testes	82
4.2	Resultados esperados	82
4.3	Resultados obtidos	83
4.3.1	Validação	83
4.3.2	Verificação	89
4.4	Considerações finais	97
5	Conclusão	98
5.1	Análise de objetivos e resultados	98
5.2	Limitações da pesquisa	99
5.3	Recomendações para futuros trabalhos	99
	REFERÊNCIAS BIBLIOGRAFICAS	101
	ANEXOS	106

LISTA DE FIGURAS

Figura 1 - Localização do projeto conceitual no PDP	12
Figura 2 - Desdobramento da função global em funções parciais e elementares. Fonte: Back et al. (2008).....	13
Figura 3 - Matriz Morfológica de soluções (ROZENFELD et al., 2006)	14
Figura 4 - Oportunidade nos estágios iniciais do projeto. Adaptado de (WANG et al., 2002).....	15
Figura 5 – Metodologias para cada etapa do processo.....	18
Figura 6 - Alargamento das restrições ao projeto para formulação do projeto conceitual. Adaptado de (BAXTER, 2000)	24
Figura 7 - Representação da Função Geral	25
Figura 8 - Ciclo do RBC, adaptado de (LOPEZ DE MANTARAS et al., 2006)	33
Figura 9 - Hierarquia das tarefas no RBC. Retirado de (AAMODT; PLAZA, 1994) ...	34
Figura 10 – Representação de casos por vetor de características, adaptado de (EL- SAPPAGH; ELMOGY, 2015)	38
Figura 11 - Arquitetura de Software (simplificado)	47
Figura 12 - Repositório para compartilhamento dos códigos gerados no projeto.....	48
Figura 13 - modelo da base de dados para armazenamento dos casos	49
Figura 14 - Representação em classes da tabela do banco de dados (UML)	51
Figura 15 - Hierarquia dos DAOs (UML)	52
Figura 16 - Modelo de persistência dos casos (UML)	53
Figura 17 – Funções do produto adaptado de (ROZENFELD et al., 2006).....	54
• Figura 18 - Áreas envolvidas na Engenharia Mecatrônica, adaptado de (ZHENG et al., 2014)	55
Figura 19 - Nível de abstração de tecnologias da Mecatrônica, adaptado de (ADAMSSON, 2005)	55
Figura 20 - Classes do projeto de controle (UML).....	60
Figura 21 – Tela de estatísticas da base de dados	63
Figura 22 - Tela de busca e intercâmbio de casos.....	64
Figura 23 - Tela de cadastro de casos	65

Figura 24 - Caixas de função completa e incompleta.....	66
Figura 25 - Editor da Matriz morfológica	67
Figura 26 - funções semelhantes para seleção de soluções.....	69
Figura 27 – Sugestão automática dos conceitos para a função "Alimentar dispositivo"	71
Figura 28 - Parametrização alternativa para a função "Alimentar dispositivo"	72
Figura 29 - Funções semelhantes a "Fornecer energia"	73
Figura 30 - Exemplos para a função "Proteger aparelho"	74
Figura 31 - Estrutura dos componentes do módulo de Visão(UML).....	77
Figura 32 - Distribuição dos valores da escala por item.....	85
Figura 33 - Conceitos sugerido para o projeto Marmita com função de aquecimento	87
Figura 34 - Matriz de soluções para o projeto de medição de volume de nível no taque de cerveja.....	88
Figura 35 - Decomposição funcional (PÉREZ; SANTOS, 2016)	90
Figura 36 - Matriz Morfológica (PÉREZ; SANTOS, 2016).....	92
Figura 37 - Matriz gerada com a base inicial (primeira parte)	93
Figura 38 - Matriz gerada com a base inicial (parte 2)	94
Figura 39 - Matriz gerada com a base final (parte 1).....	95
Figura 40 - Matriz gerada com a base final (parte 2).....	96

LISTA DE TABELAS

Tabela 1 - Componentes de algoritmos de aprendizado, Fonte (DOMINGOS 2012).	30
Tabela 2 - Frameworks considerados para o desenvolvimento.	45
Tabela 3 - Detalhamento do escopo disciplinar das funções	56
Tabela 4 - Resultados da avaliação em sala de aula	83
Tabela 5 - Projetos dos grupos avaliados	84
Tabela 6 - Resultados dos testes com os casos de uso	89

LISTA DE SÍMBOLOS

Subscritos

sim similaridade

Siglas

IA	Inteligência Artificial
RBC	Raciocínio Baseado em Casos
CAD	Computer Aided Design
CRUD	Create, Read, Update e Delete
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
QFD	Quality Function Deployment (Desdobramento da Função Qualidade)
TRIZ	<i>teoriya resheniya izobretatelskikh zadatch</i> (teoria da resolução inventiva de problemas)
FMEA	<i>Failure Mode Effects Analysis</i> (Análise de Modo e Efeito de Falha)
IDE	Integrated Development Environment
MVC	Model, View and Controller
JPA	Java Persistence API
DAO	Data Access Object
VO	Value Object

1 INTRODUÇÃO

Esta pesquisa tem como contexto a possibilidade de melhoria de uma das etapas do Processo de Desenvolvimento de Produtos: a fase de Projeto conceitual.

A fase de projeto conceitual no processo de projeto de produtos tem por objetivo a partir de um conjunto de especificações geradas, criar diferentes alternativas de soluções. (ROZENFELD et al., 2006). Esta compreendida entre as fases de projeto Informacional e a Fase de Projeto Detalhado, ilustrada na Figura 1.

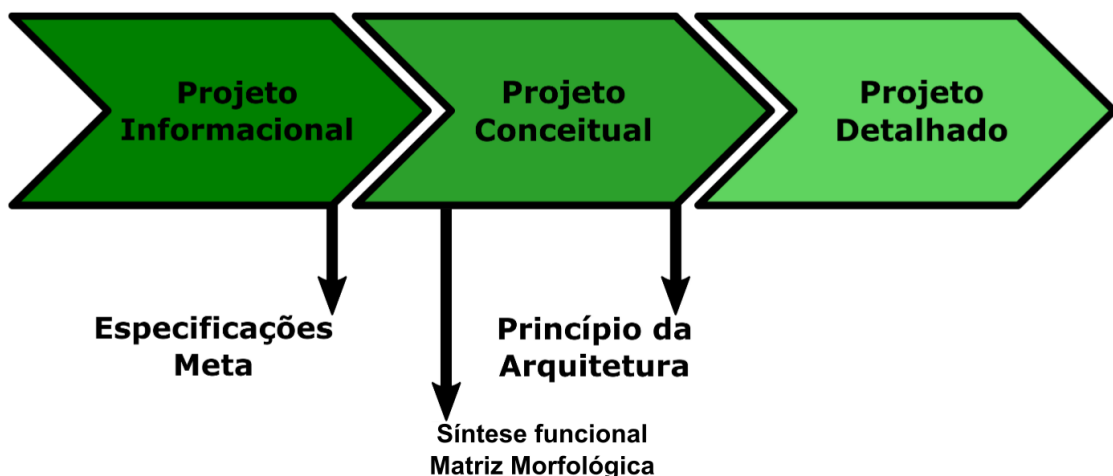


Figura 1 - Localização do projeto conceitual no PDP

(LOPEZ FLORES et al., 2015) argumentam que a fase de projeto conceitual é uma das mais intensas de métodos de criatividade e difícil de ser alcançada. Esta se caracteriza então como fase de criatividade, na qual é empregado um conjunto de métodos e técnicas para estimular a criação de um maior número de alternativas de solução que possam resolver o problema de projeto. Entre os diferentes métodos de criatividade podem ser citados os métodos de

criatividade sistemáticos, tais como o método da síntese funcional e o emprego da matriz morfológica.

O método de Síntese Funcional é o mais apropriado ao desenvolvimento de sistemas técnicos que: são uma integração complexa de elementos e que serve para um propósito determinado; o propósito é declarado a partir da formulação da função global do sistema; a função global é desdobrada, sucessivamente, em funções elementares ou parciais (Figura 2); são considerados como processos de transformação de estado e das propriedades de grandezas do tipo de energia, material e sinal (BACK, N.; OGLIARI, A.; DIAS, A.; SILVA, 2008)

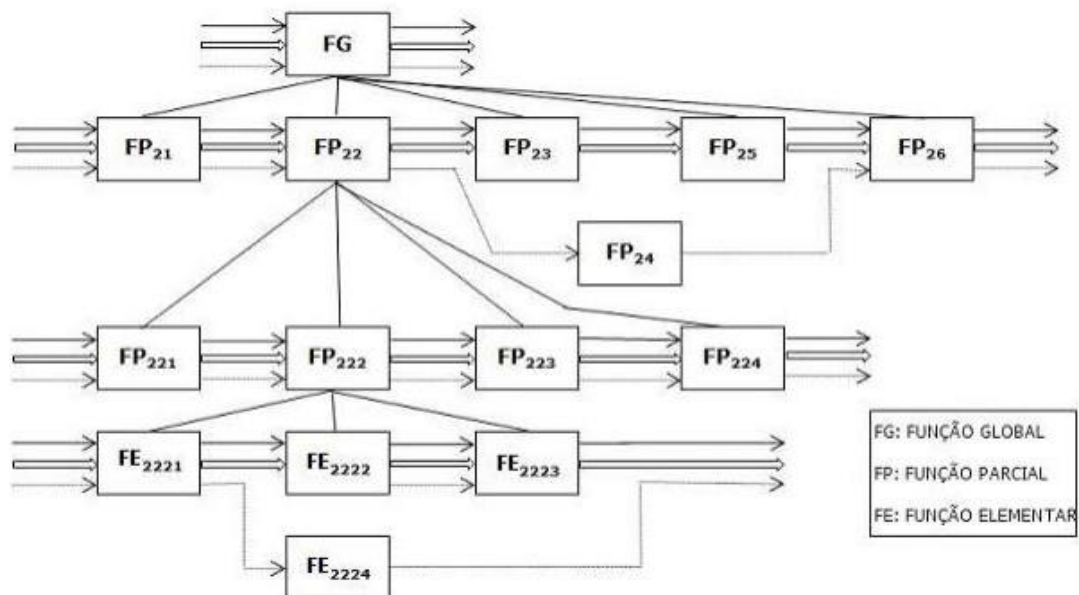


Figura 2 - Desdobramento da função global em funções parciais e elementares. Fonte: Back et al. (2008).

O método da matriz morfológica tem por objetivo encontrar uma nova solução do problema partindo de uma pesquisa sistemática de diferentes combinações de elementos ou parâmetros. As combinações desses elementos servirão de inspiração para geração de novas soluções para os problemas, como mostrado na Figura 3.

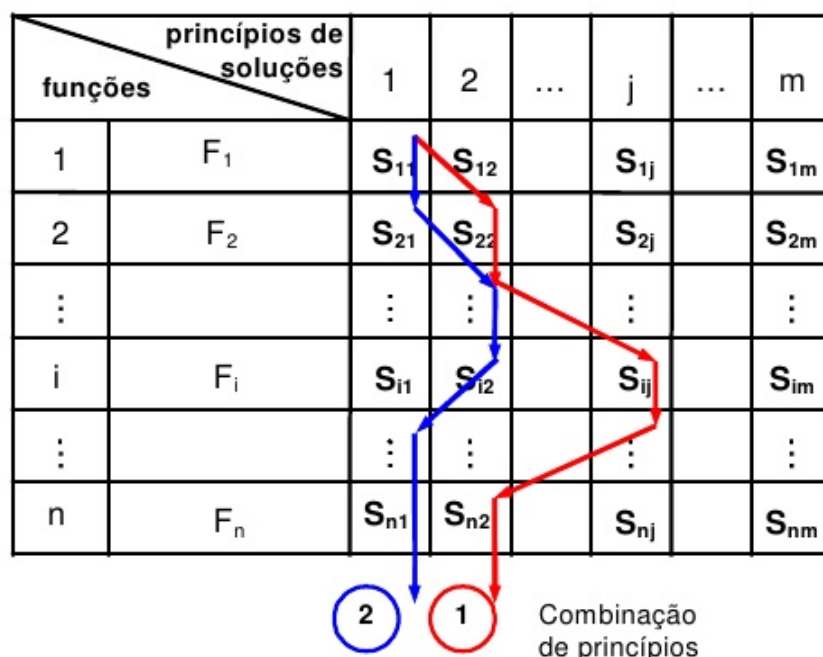


Figura 3 - Matriz Morfológica de soluções (ROZENFELD et al., 2006)

O resultado desta fase será um princípio de arquitetura que será avaliado nas próximas fases do projeto para que desta surja o que será a estrutura final do produto e desenvolvida pela equipe técnica. As decisões tomadas neste momento terão ramificações profundas no resto do projeto, tornando crítica a fase de Projeto Conceitual.

Decisões tomadas durante a fase de projeto conceitual têm influência significativa no produto final. Estima-se que mais de 75% do custo do produto final seja resolvido na fase conceitual. Foram propostas e aplicadas diferentes metodologias para o projeto detalhado do produto, mas mesmo o mais alto padrão de projeto detalhado não pode compensar um conceito de projeto falho formulado na fase de projeto conceitual (SELL; TAMRE, 2016).

(WANG et al., 2002) argumentam que mesmo com a fase de projeto conceitual ter um alto impacto para o projeto, esta fase ainda apresenta um pequeno grupo de ferramentas de apoio ao processo de projeto, conforme ilustrado na Figura 4.

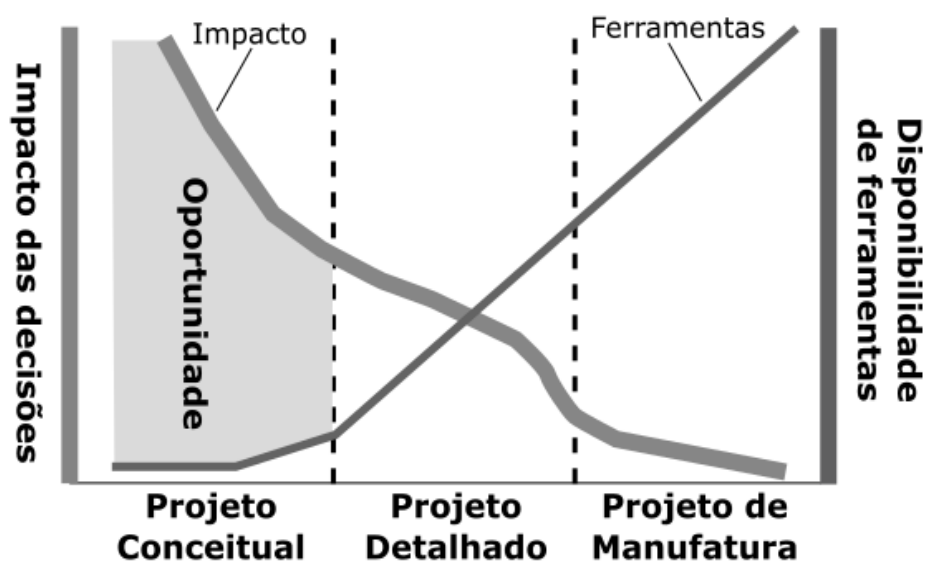


Figura 4 - Oportunidade nos estágios iniciais do projeto. Adaptado de (WANG et al., 2002)

(CHANDRASEGARAN et al., 2013) corroboram que o emprego de ferramentas computacionais auxilia na melhoria da eficiência e eficácia do processo de projeto de produto, pois são capazes de reter as informações acumuladas, capturando o conhecimento produzido ao longo do processo de projeto e permitindo o acesso no momento mais oportuno.

(CATECATI et al., 2011) colocam que o processo de projeto se modificou de forma drástica durante as últimas décadas, por meio da introdução e rápida popularização da utilização de microcomputadores em ambientes de projeto (sistemas CAD, CAE, CAM, entre outros) e com a introdução de novos métodos e ferramentas de projeto (DFX, QFD, TRIZ, FMEA, entre outros), reduzindo o tempo de desenvolvimento dos produtos, dos seus custos associados e, ainda assim, a um aumento da qualidade final dos produtos desenvolvidos.

Sistematizar o conhecimento para a recuperação das soluções é algo que já vem sendo tentado como forma de melhorar o desempenho dos projetistas, em especial em projetos que requerem um gerenciamento de uma grande quantidade de informações, como é o caso dos projetos multidisciplinares.

O desenvolvimento de ferramentas de suporte à fase de projeto conceitual, pode trazer benefícios para resgate do conhecimento de princípios de solução,

contribuindo com a geração de novas soluções bem como para o desempenho do processo de projeto de produtos. Tarefas que envolvem exploração de conceitos em domínios multidisciplinares e que podem trazer grande dificuldade para projetistas humanos, que possuem formação somente sobre certas disciplinas. Sendo assim, uma ferramenta de Projeto Conceitual Assistido por Computador (PCAC) poderia ser desenvolvida para ajudar projetistas humanos a alcançar o Projeto Conceitual Criativo (PCC) em domínios multidisciplinares (CHEN; LIU; XIE, 2012).

Sistemas de aprendizado baseados em Aprendizado de máquina e que se utilizam de Raciocínio Baseado em Casos (ou *Case Based Reasoning*) já são utilizados em várias áreas como ferramentas de sugestão de soluções. (LOPEZ FLORES et al., 2015), por exemplo, propõe uma ferramenta baseada em RBC em conjunto com ferramentas provenientes do TRIZ para prover uma metodologia de resolução para a etapa do projeto conceitual. Tais sistemas conseguiriam prover tanto informações valiosas de projeto quanto através delas ajudar na solução de problemas relacionados a etapa conceitual do projeto de produto.

O uso de novas tecnologias de informação, tais como Web Semântica, *data mining*, *text mining*, teoria do caos e algoritmos evolucionários estão sendo utilizados para prever com maior precisão os próximos passos de evolução tecnológica e no desenvolvimento de novos produtos (LEON, 2009).

Esta pesquisa tenta entender como pode ser melhorada a etapa de Projeto conceitual de produtos através do uso de métodos de aprendizado de máquina, como forma de gerenciar e sugerir soluções para o usuário que estejam dentro do contexto do projeto executado, facilitando o desenvolvimento e melhorando a qualidade da arquitetura inicial gerada.

1.1 Objetivo da pesquisa

Propor o emprego do aprendizado de máquinas como ferramenta de apoio a fase de projeto conceitual para melhoria de eficiência e eficácia do processo de projeto de produtos mecatrônicos.

1.1.1 Objetivos Específicos

Pretende-se alcançar o objetivo da pesquisa através da execução das seguintes tarefas:

- Documentar e analisar casos de desenvolvimento de Produtos Mecatrônicos, mais especificamente em sua fase conceitual.
- Criar uma base de dados de fases conceituais de Projetos Mecatrônicos já finalizados para o uso no sistema de aprendizado
- Criar um sistema de Aprendizado capaz de utilizar os casos cadastrados para uso no auxílio da fase de projeto conceitual.
- Criar uma interface que possa ser utilizada em meio acadêmico como forma de agregar conhecimento a base construída
- Validar o sistema através de seu uso no âmbito de projeto de produtos

1.2 Metodologia da pesquisa

As fases nas quais a pesquisa foi dividida são ilustradas na Figura 5. Resumidamente, as fases foram divididas em: avaliação dos problemas disponíveis para pesquisa, seu contexto e aplicabilidade, relacionar autores e pesquisas na área provendo base de informação para a solução envisionada e finalmente, execução do projeto tendo como base as informações levantadas nas etapas anteriores.

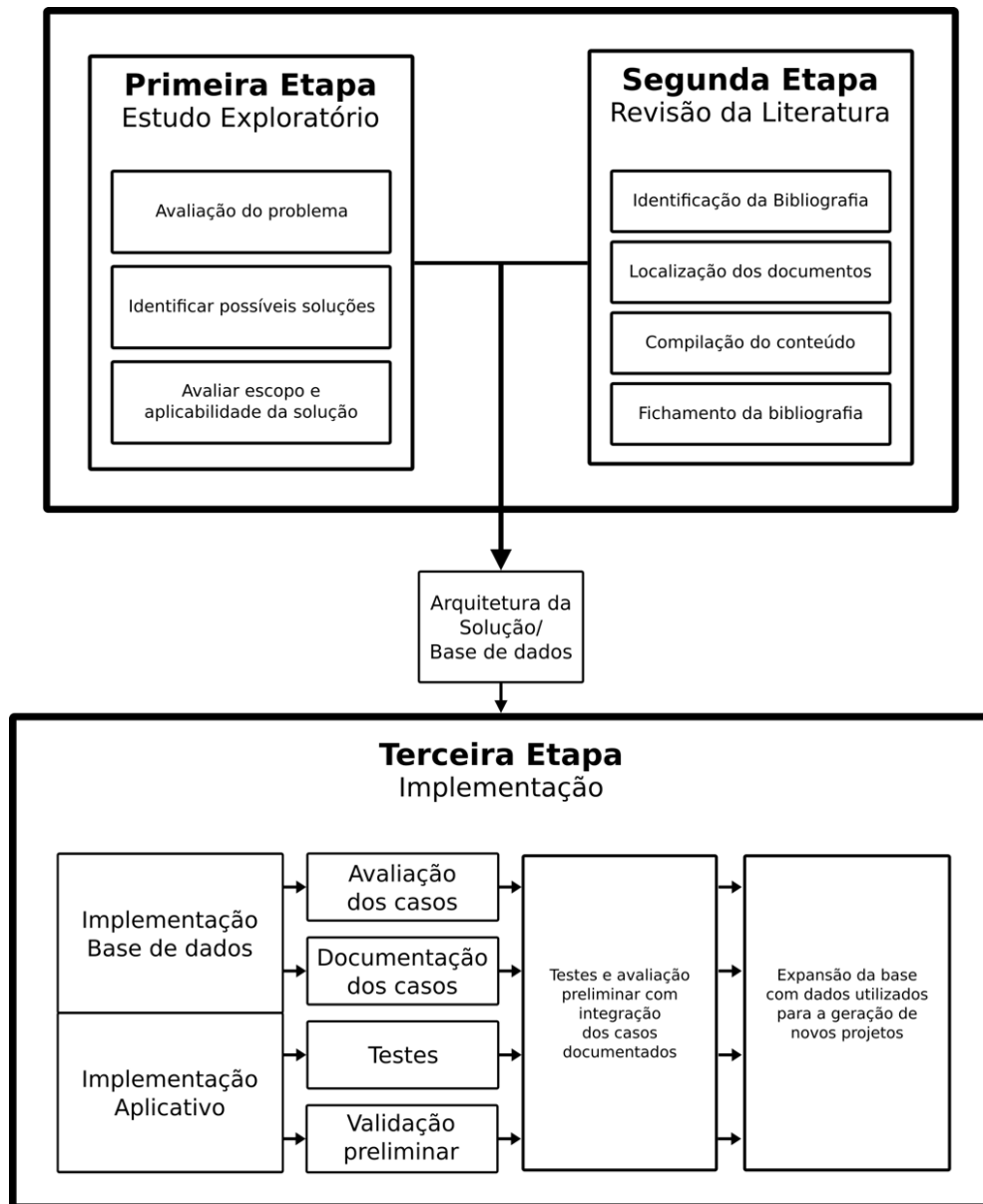


Figura 5 – Metodologias para cada etapa do processo.

- Estudo exploratório
 - Busca na literatura por problemática semelhante a proposta para refinamento do objetivo de pesquisa.
 - Busca por aplicações de sistemas baseados em RBC para contextualização.
 - Busca por ferramentas de auxílio computacional ao PDP já desenvolvidas.

- Busca por sistemas de sugestão já desenvolvidos para auxílio no PDP.
- Busca por formas de avaliação do sistema como ferramenta de software e de projeto.
- Revisão de Literatura
 - Pesquisa dos critérios de avaliação e criação dos conceitos para a fase de projeto conceitual do PDP.
 - Pesquisa sobre os sistemas de aprendizado de máquina baseados em RBC.
- Sistema de aprendizado – A metodologia aqui seguida para o desenvolvimento do sistema é um modelo de trabalho para desenvolvimento de sistemas de aprendizado de máquina, sugerido por (BROWNLEE, 2014).
 - Definir o problema
 - Preparar os dados
 - Selecionar e verificar os algoritmos
 - Melhorar resultados
 - Apresentar resultados

Com o problema já definido, as outras etapas que se seguem serão divididas em duas frentes: A captação dos dados, que serão em um primeiro momento recuperados a partir de projetos acadêmicos fornecidos pelo departamento de Engenharia Mecatrônica da Universidade de Brasília e o desenvolvimento das funcionalidades de aprendizado de máquina.

- Análise e documentação dos casos.
 - Os casos serão analisados de acordo com sua relevância para a pesquisa e deverão ser introduzidos ao conjunto de casos

utilizáveis separando-os em casos de treinamento e casos de validação

- Validação do sistema.
 - Para validar de teste piloto e observação de pontos de erro e de melhoria fora do ambiente de desenvolvimento. Esta avaliação será feita de duas formas, em acordo com (FDA, 2002):
 - Verificação do software: A ferramenta será inserida no ambiente de testes para verificar o seu uso por um grupo controlado de usuários.
 - Validação do software: Serão gerados casos de uso para avaliar o desempenho da ferramenta. Serão avaliados os resultados esperado em detrimento dos resultados obtidos para cada caso de uso.

1.3 Estrutura do trabalho

A estrutura do trabalho segue a listagem abaixo, aonde são explicitados os pontos abordados em cada um dos capítulos:

- **Capítulo 1: Introdução** – Contextualização do problema, metodologias utilizadas e objetivos a serem alcançados durante a pesquisa.
- **Capítulo 2: Revisão Bibliográfica** – Revisão do estado da arte dos assuntos abordados pela pesquisa, conceitos e técnicas relevantes.
- **Capítulo 3: Desenvolvimento do sistema** – Uma visão geral do processo de desenvolvimento do sistema, desde a decisão da arquitetura do software até a finalização do mesmo.
- **Capítulo 4: Resultados alcançados** – Análise dos resultados obtidos com a utilização do software e uma amostra das soluções por este geradas.

- **Capítulo 5: Conclusões e sugestões para trabalhos futuros** – Algumas considerações sobre melhorias e as conclusões sobre o trabalho desenvolvido.

Em sequencia será mostrada a base teórica estudada para a realização do trabalho e o contexto no qual o problema escolhido será abordado.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados os conceitos teóricos envolvidos na elaboração do trabalho aqui desenvolvido. O contexto dentro do qual o trabalho se encontra é o desenvolvimento de Produtos Mecatrônicos e a Tecnologia da Informação. Será abordada a maneira como ambas podem ser integradas para gerar a ferramenta computacional proposta em cinco sessões.

2.1 O Projeto na Engenharia Mecatrônica

O projeto na Engenharia mecatrônica envolve conhecimentos de várias disciplinas, ou seja, multidisciplinares, então a quantidade de conhecimento necessário para desenvolver tais produtos/sistemas é ampla.

Uma das principais dificuldades no projeto na Engenharia Mecatrônica é a carência de métodos e técnicas que promovam a integração entre as diferentes áreas de conhecimento. Tradicionalmente Engenheiros Mecânicos desenvolvem o seu projeto com ênfase no domínio geométrico em contrapartida os Engenheiros Eletricistas buscam soluções envolvendo um único domínio de conhecimento (HEHENBERGER; ZEMAN, 2007). Logo, a qualidade do projeto depende diretamente da qualidade da interação das equipes que executam o projeto, dos seus respectivos subsistemas, e da forma como elas irão se integrar para formar o produto final.

A sinergia técnica de um Sistema Mecatrônico cria dependências críticas entre as disciplinas de engenharia envolvidas. Estas dependências são demonstradas de muitas formas, propriedades mecânicas, por exemplo, podem ser ligadas a características do sistema de controle que por sua vez estarão ligadas a propriedades do software e vice-versa. Como resultado, dependências organizacionais se tornam de gerenciamento crítico e a cooperação entre engenheiros representando diferentes disciplinas e funções

se tornam um fator cada vez mais importante a ser considerados em organizações envolvidas com o projeto de Sistemas Mecatrônicos. (ADAMSSON, 2005).

O foco deste trabalho será entender especificamente a fase de projeto conceitual do PDP para que possamos avaliar a melhor forma de trabalhar a sua automação em especial na área de projetos multidisciplinares, como o caso da mecatrônica. Sendo assim trataremos de suas particularidades na sessão que se segue.

2.2 A Fase de Projeto conceitual

A fase de projeto conceitual é a fase que inicia a busca pelas soluções que serão desenvolvidas para o desenvolvimento do produto final. A importância desta fase é ter a responsabilidade de gerar a primeira abstração da arquitetura do produto, descritas por suas formas e funções.

Na definição de alguns autores:

(ROZENFELD et al., 2006) O projeto conceitual é a fase de projeto em que a equipe de projeto, a partir dos requisitos dos clientes e das especificações meta do produto, cria uma concepção para o produto a qual deve atender da melhor maneira possível estar especificações, sujeita as limitações de recursos e as restrições de projeto.

(RZEVSKI, 2003) Projeto conceitual é o estágio do projeto aonde os projetistas selecionam conceitos que serão empregados na solução de um determinado problema de projeto e decidem como interconectar estes conceitos em uma arquitetura de sistema apropriada.

(BAXTER, 2000) O projeto conceitual tem o objetivo de produzir princípios de projeto para o novo produto. Ele deve ser suficiente para satisfazer as exigências do consumidor e diferenciar o novo produto de outros produtos existentes no mercado. Especificamente, o Projeto Conceitual deve mostrar como o novo produto será feito para atingir os benefícios básicos, ilustrado na Figura 6. Portanto, para o projeto conceitual é necessário que o benefício básico esteja bem definido e se tenha uma boa compreensão das

necessidades do consumidor e dos produtos concorrentes. Com base nessas informações, o projeto conceitual fixa uma série de princípios sobre o funcionamento do produto e os princípios de estilo.

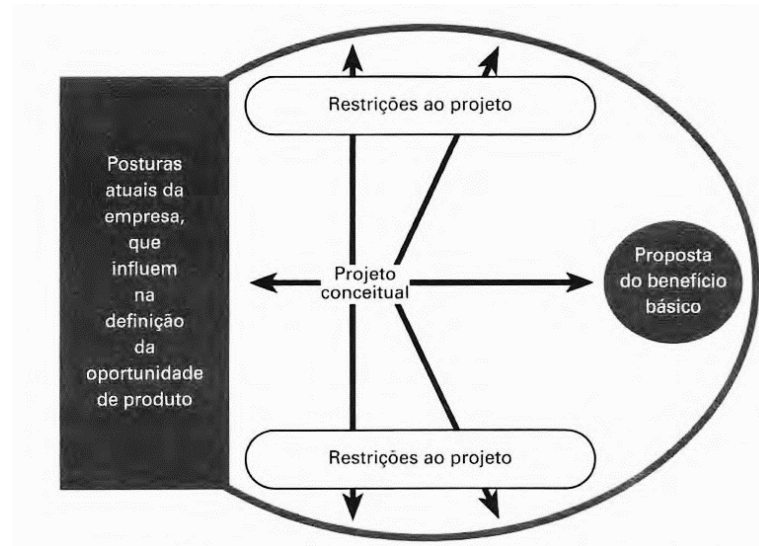


Figura 6 - Alargamento das restrições ao projeto para formulação do projeto conceitual. Adaptado de (BAXTER, 2000)

Os princípios funcionais do produto devem ser descritos de forma detalhada, e um dos meios para alcançar esta descrição é a utilização da metodologia de modelagem funcional, que consiste do desdobramento da função principal e suas derivadas até que uma lista de funções capaz de descrever o funcionamento detalhado do produto seja gerada. A cada uma destas funções são atribuídos princípios de solução que serão avaliados posteriormente de acordo com as especificações propostas inicialmente, de forma que o conjunto de soluções escolhido seja o que melhor atende as necessidades levantadas.

A modelagem funcional deve ser no maior nível de abstração, definindo de modo geral as questões de projeto evitando que os projetistas se envolvam demais com soluções que já são particulares de sua experiência. Deverá ser então a primeira etapa a ser realizada, iniciando-se com a definição da função total ou global do produto, sendo representada pela Figura 7:



Figura 7 - Representação da Função Geral

A função geral é a maneira de se representar da forma mais abstrata possível qual será a tarefa a ser cumprida pelo produto. A descrição da tarefa é dada por um par verbo + objeto que descreve a tarefa gramaticalmente, acompanhada por um trio de entradas e um trio de saídas que indica as transformações efetuadas pela função em termos de Energia, Material e Sinal.

Um exemplo bastante simples de uma função geral seria “Lavar roupas”, cujas transformações poderiam ser descritas como:

- Energia → Elétrica -> Mecânica
- Material → Roupa Suja e água limpa -> Roupa limpa e água suja
- Sinal → Programação da máquina

Este exemplo possui apenas um nível de desdobramento, porém de acordo com o projeto a profundidade poderá ser aumentada enquanto necessário até que se tenha uma descrição satisfatória para as funcionalidades do produto.

Esta análise é aplicada tanto à função geral quanto aos seus desdobramentos, de forma que as entradas e saídas do sistema desdobrado corresponda às da função geral. O resultado deste processo servirá como base para o trabalho das soluções que serão avaliadas para a geração da arquitetura resultante da fase de projeto conceitual.

(BAXTER, 2000) avalia que o sucesso do projeto conceitual se baseia na geração da maior quantidade possível de conceitos e na correta avaliação para a seleção dos melhores dentre estes. Também explicita que a tarefa de projetar conceitualmente um produto, assemelha-se a metodologia criativa em suas etapas:

- Definição dos objetivos do projeto conceitual: É dependente do tipo de produto realizada com base nas informações já levantadas acerca das necessidades que serão atendidas pelo mesmo, relacionadas ao escopo do problema e as suas restrições. Aqui são priorizadas as necessidades do consumidor em detrimento da possibilidade de fabricação do produto.
- Geração dos possíveis conceitos: Exige que seja utilizada a imaginação, intuição e raciocínio lógico para a concepção de conceitos originais que consigam suprir a função a ele associada. Pode-se citar como exemplos de técnicas:
 - Análise de tarefas – envolve a análise da ergonomia e antropometria para entender como é executada a tarefa pelo público para o qual se quer desenvolver o novo produto ou desenvolver a tarefa.
 - Análise das funções – é orientada ao consumidor. Tem como objetivo mostrar à equipe de projeto como o produto deve ser projetado.
 - Análise do ciclo de vida – tem como foco a diminuição do impacto de um produto no meio ambiente.

Estes métodos são úteis para a redução dos problemas aos seus elementos básicos e analisar diferentes aspectos do projeto para gerar o maior número de soluções alternativas.

- Seleção de um conceito de acordo com a especificação do projeto: a seleção do conceito que será desenvolvido no projeto é uma tarefa iterativa, que exige uso de criatividade, e não somente a análise lógica para escolha do melhor conceito. Existem métodos como o de convergência controlada, que envolvem a combinação de vários dos conceitos gerados.

Uma das maneiras de associar as funções aos conceitos que serão responsáveis pelo cumprimento da tarefa a elas associada é a matriz morfológica.

A matriz morfológica é uma metodologia tradicional de seleção de conceitos, permitindo ao projetista a seleção dos melhores conceitos de projeto através da combinação de maneira sistemática dos vários “atuadores” que poderiam satisfazer cada função parcial. Algumas ressalvas, porém são listadas pelo autor a respeito do uso deste método (FARGNOLI; ROVIDA; TROISI, 2006):

- Não fornecem muitas informações quanto à viabilidade das soluções informadas.
- Por vezes leva ao “exagero” no projeto, impactando na efetividade das tarefas de solução de problemas.
- Requer um time de especialistas para aumentar a possibilidade de uma solução inovadora.

A partir dela então se inicia a discussão sobre a arquitetura do produto. O conjunto de soluções nela listado deverá ser utilizado para compor uma solução inicial para o projeto.

Uma importante ferramenta para a combinação de princípios de solução individuais em princípios de solução totais para o produto é a matriz morfológica. A matriz morfológica dispõe simultaneamente as funções que compõem a estrutura funcional escolhida para o produto e as diversas possibilidades de solução para elas. A matriz morfológica possibilita uma análise das possíveis configurações para o produto projetado (ROZENFELD et al., 2006).

A geração dos conceitos relacionados à fase de criatividade é um trabalho que depende do conhecimento e capacidade cognitiva e criativa dos projetistas. Por este motivo uma das tecnologias pensada para o uso em conjunto com as metodologias acima descritas foi o aprendizado de máquina, que tem como preceito a imitação das funções cognitivas dos seres humanos.

2.3 Aprendizado de máquina

Aprendizado de máquina ou *Machine Learning* é uma área da Ciência da computação que estuda a utilização de sistema para o reconhecimento avançado de padrões em conjuntos de dados.

A “programação” destes sistemas é na verdade o seu treinamento, utilizando um conjunto de dados que servirá para que a rede gere um modelo próprio capaz de reconhecer em outro conjunto de dados com as características para a qual foi treinado.

De forma mais sucinta (BROWNLEE, 2013) colocam que o aprendizado de máquina é o treinamento de um modelo a partir de dados que generalizam uma decisão em detrimento de uma medida de desempenho.

2.3.1 Conceitos e utilização

O aprendizado de máquina é um dos campos dentro de outro chamado *Data Mining*. Entender o que é este conceito envolve entender do que se trata o aprendizado e como um sistema pode ser capaz de entender e associar padrões previamente encontrados a novos que são semelhantes, ou de acordo com (WITTEN; FRANK; HALL, 2011) Coisas aprendem quando elas mudam o seu comportamento de forma com que seu desempenho melhore futuramente.

O aprendizado de máquina tem suas bases na junção entre a metodologia estatística e na computação. (MITCHELL, 2006) coloca que enquanto a estatística se foca nas informações que podem ser recuperadas de conjuntos de dados, a computação adiciona a preocupação com a captura, armazenamento, indexação, recuperação e junção destes dados e como as subtarefas de aprendizado podem ser coordenadas em sistemas maiores e questões de tratabilidade.

(DENG, 2013) argumenta que historicamente o conceito de *Deep Learning* se originou do estudo de redes neurais. Redes de Alimentação progressiva com muitas camadas ocultas são um bom exemplo de modelos com esta arquitetura.

(BAQAPURI, 2015) justifica que um aspecto importante do *Deep Learning* é usar sistemas automaticamente treináveis fim-a-fim que não dependem de heurísticas desenvolvidas por humanos. Sistemas de aprendizado de máquina tradicionais são divididos em dois módulos. Primeiramente um extrator de características para modulação dos dados de entrada em vetores de poucas dimensões que podem ser facilmente comparados e avaliados e que são relativamente invariantes a distorções. Estes dados são então utilizados para alimentar um módulo classificador que é de uso geral e treinável. Duas desvantagens deste tipo de arquitetura são que o desempenho é ditado por entradas humanas e o extrator de características é específico da tarefa, tendo de ser reescrito para cada tarefa.

Um dos maiores usos desta disciplina é o reconhecimento de padrões. Vários tipos de sistemas se prestam a realizar o reconhecimento de padrões de imagem, podendo inferir destes dados informações que até antes somente os seres humanos eram capazes.

(MITCHELL, 2006) cita como exemplos de utilização:

- Reconhecimento de fala – sistemas de reconhecimento de padrões sonoros podem ser utilizados para reconhecer a fala humana e traduzi-la em texto para uso em outras atividades.
- Visão computacional – Pode ser utilizada para reconhecimento facial, classificação de imagens, reconhecimento de escrita cursiva.
- Vigilância Biológica – Padrões de disseminação de doenças podem ser detectados e rastreados utilizando aprendizado de máquina.
- Controle robótico – Utiliza-se como sistema de auxílio a navegação para reconhecimento e desvio de obstáculos
- Aceleração de ciências empíricas – Podem ser utilizados para o aprendizado de modelos e análise de padrões complexos para uso científico

Os algoritmos utilizados para o processamento dos sistemas são variados, e cada um possui particularidades que devem ser analisadas previamente ao

desenvolvimento do sistema para que o escolhido consiga gerar o resultado ótimo para o tipo de problema apresentado. (DOMINGOS, 2012) cita três parâmetros a serem analisados para a correta escolha do algoritmo:

- Representação – Um classificador deve ser representado em alguma linguagem formal que o computador possa lidar. Assim, a escolha de uma representação é equivalente a escolha de um conjunto de classificadores que o classificador pode aprender. Isto é chamado de espaço de hipótese de um sistema de aprendizado.
- Avaliação – Uma função de avaliação é necessária para distinguir bons classificadores de ruins. A função de avaliação usada internamente pelo algoritmo pode diferir da externa que queremos otimizada pelo classificador.
- Otimização – Finalmente, é necessário um método para buscar dentre os classificadores da linguagem pelo melhor. A escolha da técnica de otimização e a chave para a eficiência do aprendizado e ajuda a determinar o classificador produzido se a função de avaliação possuir mais de um ótimo.

Exemplos de cada um dos algoritmos para cada etapa são listados na Tabela 1, mostrada logo abaixo:

Tabela 1 - Componentes de algoritmos de aprendizado, Fonte (DOMINGOS 2012).

Representação	Avaliação	Otimização
Instances	Taxa de	Otimização
K-Vizinho mais	Erro/Acerto	combinatória
próximo	Precisão e	Pesquisa ávida
Máquina de	revocação	<i>Beam search</i>
vetores de suporte	Erro Quadrático	<i>Branch-and-bound</i>
Hiperplanos	Probabilidade	Otimização
Naive Bayes	Probabilidade a	continua

Regressão	Posteriori	Otimização sem
logística	Ganho de	Restrições
Árvores de	Informação	Gradiente
decisão	Divergencia de	descendente
Conjuntos de	Kullback-Leibler	Gradiente
regras	Custo / Utilidade	conjugado
Lógica		Métodos Quasi-
Proposicional		Newton
Redes Neurais		Programação
Programação		Linear
Linear		Programação
Modelos gráficos		Quadrática
Redes bayesianas		
Campos aleatórios		
condicionais		

(DOMINGOS, 2012) Argumenta que o fundamento do aprendizado de máquina é generalizar além dos exemplos com os quais foi treinado. Isto se deve ao fato de que não importando a quantidade de dados na sua base de treinamento, é improvável que os mesmos apareçam novamente exatamente da mesma forma quando o sistema for testado.

Isto gera a importância de além de se ter dados para o aprendizado, ter-se também um conjunto de dados que possam ser utilizados exclusivamente para testes.

(WITTEN; FRANK; HALL, 2011) Uma maneira de visualizar o problema do aprendizado é imaginar uma busca através de possíveis descrições de conceitos por uma que se encaixe nos dados então o processo de

generalização pode ser entendido como uma busca por um enorme, mas finito conjunto de dados.

2.3.2 Raciocínio Baseados em Casos

O Raciocínio Baseado em Casos (RBC) consiste em uma maneira de processar informações sobre soluções de problemas anteriores, em um sistema de Inteligência artificial, de forma a propor com base nestas informações, novas soluções para um novo problema. Cada caso solucionado então pode passar a fazer parte da base de conhecimento para uso em novas soluções.

(RICHTER; AAMODT, 2005) colocam que o Raciocínio baseado em casos envolve a captura e utilização de experiências específicas. A maior influência da ciência cognitiva em RBC é centrada nos termos experiência, memória e analogia.

É um modelo também baseado em aspectos da cognição humana, no qual vários casos anteriores podem ser utilizados como conhecimento para a geração de novos conceitos que podem ser aplicados a outros problemas.

(RICHTER; AAMODT, 2005) argumentam que o RBC tem laços profundos com aprendizado de máquina, apesar de não poder ser considerado uma de suas subáreas. A aproximação utilizada pelo RBC é conhecida como *lazy learning*.

O aprendizado em RBC vem da experiência adquirida da solução dos casos, reutilizando casos resolvidos satisfatoriamente como base para a solução de novos, deixando de lado a generalização do problema proposta pelas vertentes mais tradicionais de *machine learning* (RICHTER; AAMODT, 2005). Como um dos principais paradigmas da IA (Inteligência Artificial), RBC representa o conhecimento e seu raciocínio. Um corpo de casos representa conhecimento de primeira classe, a partir dos quais os métodos de raciocínio para busca de similaridades, adaptação de casos e novos aprendizados são aplicados.

(LOPEZ DE MANTARAS et al., 2006) O ciclo do aprendizado no RBC consiste na passagem pelas seguintes etapas, ilustrado na Figura 8:

- Recuperação – busca de um conjunto de casos similares, comparados em relação ao um conjunto de características escolhidas para representação da informação nele contida.
- Reuso – os casos da base são então reutilizados para a geração da solução para o novo problema proposto.
- Revisão - avaliação da solução gerada para atestar sua qualidade
- Retenção – uma solução satisfatória pode então passar a integrar a base como um novo caso.

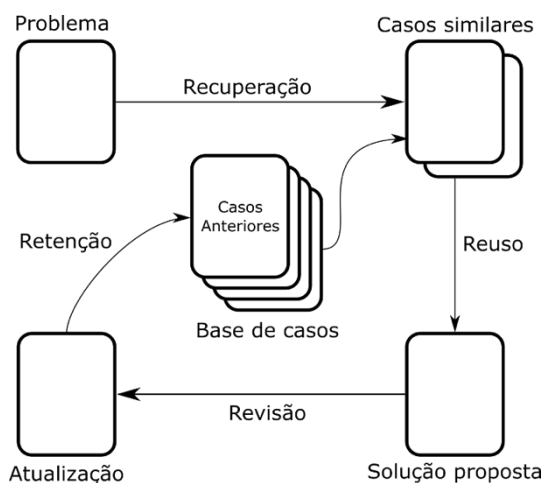


Figura 8 - Ciclo do RBC, adaptado de (LOPEZ DE MANTARAS et al., 2006)

Ou como dito em (RICHTER; AAMODT, 2005) Um novo problema é resolvido através da RECUPERAÇÃO de um ou mais casos previamente experimentados, REUTILIZANDO o caso de uma maneira ou outra, REVISANDO a solução baseada em outro caso e RETENDO a nova experiência agregando o caso solucionado à base de conhecimento.

Cada um dos passos acima citados possui outro conjunto de tarefas e subtarefas necessárias para a realização do ciclo completo, como mostrado ilustrado na Figura 9:

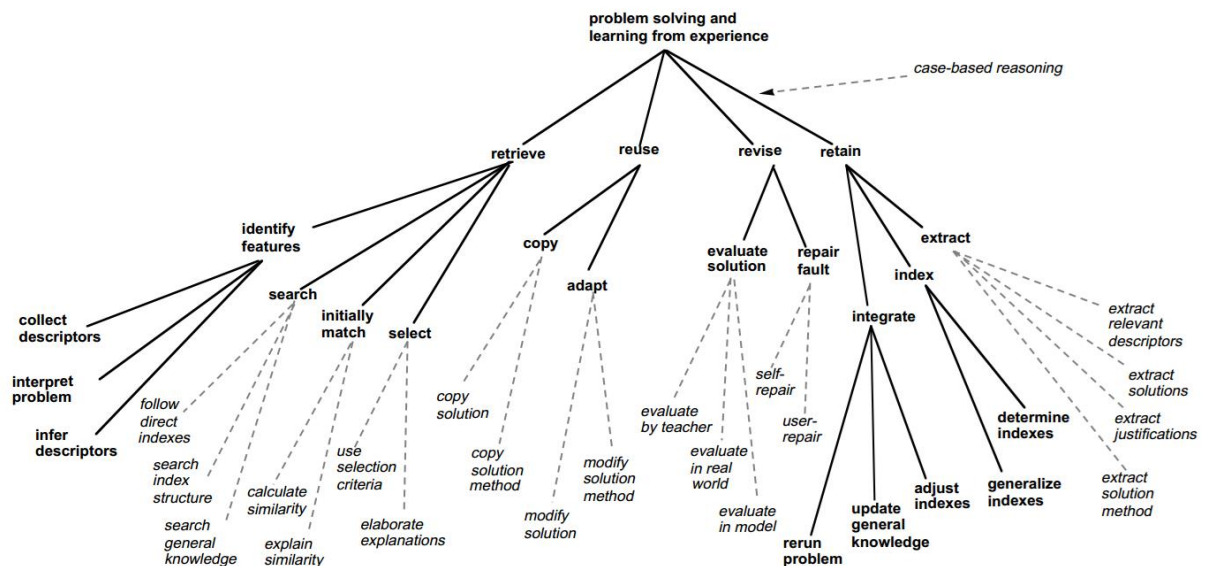


Figura 9 - Hierarquia das tarefas no RBC. Retirado de (AAMODT; PLAZA, 1994)

Apesar da estrutura aqui apresentada, a literatura deixa claro que para cada caso é necessária uma avaliação de quais tarefas e métodos serão utilizados para o alcance do resultado desejado.

(AAMODT; PLAZA, 1994) Assim como para IA em geral, não existe métodos universais para cada domínio de aplicação. O desafio em RBC é desvendar os métodos apropriados para a solução de problemas e aprendizado em domínios específicos e em ambientes de aplicação particulares. Alinhado com o modelo de tarefas mostrado, os problemas centrais atacados pela pesquisa em RBC são divididos em quatro áreas:

- Representação de conhecimento
- Métodos de recuperação
- Métodos de reuso
- Métodos de revisão
- Métodos de retenção

Assim, alinhado com o ciclo do RBC mostrado anteriormente, cada uma destas áreas ataca uma de suas principais tarefas. Na próxima seção veremos com mais detalhes as principais metodologias para cada uma das etapas listadas.

2.3.3 Principais métodos e paradigmas para RBC

(LOPEZ DE MANTARAS et al., 2006) Listam alguns métodos para cada uma das etapas do RBC:

- Recuperação - Inicialmente, o processo de geração de novas soluções passa pela etapa de recuperação dos casos similares para que os mesmos possam contribuir para a nova solução dentro do contexto necessário. Os fatores que serão utilizados para comparação são então cruciais para que os casos escolhidos estejam realmente no mesmo domínio do problema a ser solucionado.
 - Avaliação de similaridade – são utilizadas medidas de similaridade superficiais, geralmente recuperada da descrição do problema e representadas utilizando pares de atributo-valor. Pode ser alcançada através de vocabulos indexados como forma de deixar mais explícita a descrição dos casos.
 - Recuperação direcionada à adaptação – nesta aproximação, juntam-se a preocupação da similaridade com a da utilidade. Os casos então não teriam que ser somente semelhantes, mas também úteis ao domínio em questão. São utilizadas então medidas locais e globais de adaptabilidade para assegurar-se que o caso mais adaptável seja selecionado.
 - Recuperação consciente de diversidade – para que os casos selecionados não sejam demasiadamente similares (o que pode gerar um problema, limitando as escolhas do usuário para a solução), esta metodologia visa inserir diversidade nos casos recuperados para diversificar as opções de solução.
 - Recuperação direcionada ao compromisso – também visa à inserção de elementos de diversidade nos casos selecionados, mas usa como base medidas de compromisso (requisitos não satisfeitos) que o usuário estaria preparado a aceitar.

- Recuperação baseada em ordenamento – deixa de lado a pontuação dos casos para, utilizando-se de uma linguagem de consulta expressiva, definir e combinar relações de ordenamento, gerando um conjunto de casos para uso que é naturalmente diversificado.
- Recuperação orientada à explicação – apresenta uma “justificativa” para a escolha dos casos (em alguns sistemas é desejável saber a razão da escolha) através da mostra dos seus vizinhos mais próximos.
- Reuso e revisão – Reuso vem logo após a recuperação dos casos, aonde os casos relevantes para a solução serão adaptados para a criação da nova solução. Nos casos mais simplificados, a nova solução pode ser simplesmente uma reaplicação da solução já recuperada, porém caso a diferença entre o problema a ser solucionado e o caso recuperado da base seja mais profunda é necessária a adaptação. Alguns métodos são exemplificados abaixo:
 - Substituição – simplesmente um processo de adaptação que substitui algumas partes da solução recuperada por novas para a geração da solução final
 - Transformação – altera a estrutura da solução
 - Adaptação generativa – avalia os casos recuperados aplicando ao problema em questão as soluções destes casos, ou seja, aplica as soluções recuperadas da base ao contexto do problema atual para avaliar os resultados e avaliando os erros e acertos de cada um para compor uma solução nova final.
- Retenção – A última etapa é a decisão da integração do caso solucionado à base de conhecimento do sistema. Uma das maiores questões desta etapa é se o problema a ser integrado é de utilidade ao sistema. Vários fatores devem ser levados em conta, desde a qualidade da solução até a presença de casos excessivamente similares na base

de informação, que podem degradar o desempenho do sistema. Devem ser levados em conta nesta etapa os seguintes aspectos:

- Extração – Os casos que serão incluídos na base podem vir de várias fontes (calculados pelo sistema ou cadastrados pelo usuário).
- Indexação - quais os tipos de índices serão utilizados para identificar os casos e recupera-los.
- Integração – compreende a inclusão do caso na base de conhecimento.

Após a retenção, pode-se avaliar o caso inserido através da inserção do problema novamente para averiguar se o mesmo se comporta da forma esperada.

Decidir as metodologias com as quais o sistema será desenvolvido é apenas parte do processo, e tão importante quanto decidir os caminhos pelos quais a informação será tratada é avaliar e decidir um formato apropriado para a informação. Na próxima sessão serão explicadas algumas metodologias de modelagem dos problemas.

Representação dos casos

(EL-SAPPAGH; ELMOGY, 2015) A representação dos casos em RBC apresenta três problemas gerais:

- Definição dos atributos do caso
- Definição da estrutura descritiva do caso
- Organização dos casos na base

Tais características influem diretamente na qualidade das soluções e no desempenho do sistema. O modelo de indexação dos casos não necessita ser atrelado ao modelo de dados, sendo preferencialmente desacoplado da arquitetura de armazenamento de dados que será utilizada.

Representação por vetor de características

Forma mais simples de representação, aonde cada caso é mostrado como um conjunto de características que descrevem o problema (atributo - valor). Isso facilita as comparações, visto que cada característica é comparada com o seu equivalente entre os novos casos e os casos da base, como mostrado na Figura 10.

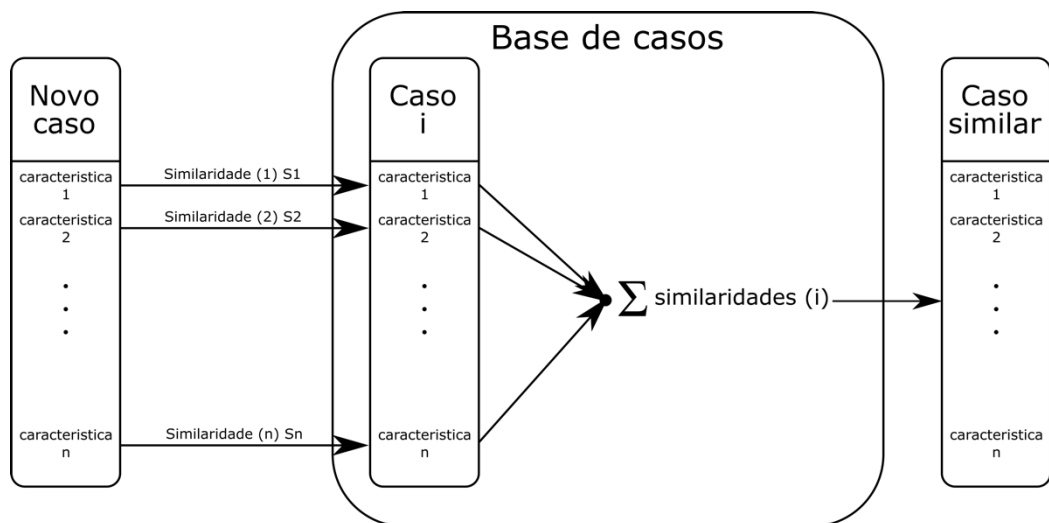


Figura 10 – Representação de casos por vetor de características, adaptado de (EL-SAPPAGH; ELMOGY, 2015)

Representação baseada em quadros

Cada caso é representado como um quadro ou *frame*, que contém o seu próprio nome e conjunto de atributos, provendo uma maneira de organizar o conhecimento em compartimentos a fim de descrever vários atributos e características do objeto. Cada espaço contém uma característica do objeto ou uma indicação para outro frame.

Representação orientada a objetos

Os casos são representados como uma coleção de objetos, cada um dos quais descritos por um conjunto de pares de atributo-valor. Este método permite a construção de casos mais complexos, sendo mais flexível do que a representação por vetores, e possui a vantagem de ser mais fácil de trabalhar o armazenamento de dados.

Representação textual

Representação onde algumas ou todas as formas de conhecimento estão em forma de texto. Várias maneiras podem ser elaboradas para este tipo de representação, como sistemas de pergunta-resposta ou vetores com pesos baseados na frequência de ocorrência das palavras ou da similaridade entre estas.

Representação hierárquica

Aplica aos casos vários níveis de abstração e detalhamento, possivelmente utilizando múltiplos vocabulários. A comparação para recuperação dos casos pode ser feita então baseada também nos níveis de abstração dos casos, permitindo assim maior refinamento das soluções.

Existem ainda mais representações possíveis, algumas com objetivos mais específicos para áreas mais específicas (existem representações utilizadas comumente para representação de casos médicos, por exemplo), confirmando que a representação dos casos depende diretamente do domínio para o qual serão aplicadas.

2.3.4 Motivação para o uso do RBC

(MAIN; DILLON; SHIU, 2001) Explicitam as razões para o uso do RBC, muitas das quais se encaixam muito bem no contexto do projeto conceitual, e são listadas abaixo de forma resumida:

- Redução das tarefas de aquisição de conhecimento – O conhecimento adquirido consiste somente na coleção de experiências relevantes e sua representação
- Evita a repetição de erros anteriores – Pode-se armazenar tanto as soluções de sucesso quanto as falhas, para serem utilizadas como referência no futuro
- Degradação graciosa de desempenho – O sistema consegue raciocinar soluções para funções onde os dados estão incompletos ou falhos.

- Possibilidade de avaliar em domínios que ainda não são completamente compreendidos, definidos ou modelados – pode funcionar mesmo com somente alguns casos no mesmo domínio do problema.
- Pode realizar predições quanto ao possível sucesso de uma possível solução – pode-se fazer esta predição baseando-se no sucesso de soluções anteriores e na sua semelhança com o contexto atual da solução atual.
- Aprendizado com o tempo – Quanto mais situações são encontradas, maior a gama de situações aonde o sistema poderá ser utilizado.
- Raciocinar em um domínio com um uma quantidade pequena de conhecimento – O RBC possibilita iniciar o sistema com uma quantidade pequena de casos e expandir o seu conhecimento na medida em que novos casos são adicionados.
- Evita refazer todos os passos necessários para alcançar uma solução – Em domínios de problemas aonde são necessários processos de trabalho significativo para alcançar a solução, modificar uma solução prévia pode reduzir o trabalho realizado.
- Provê meios de explicação – os casos prévios podem ser utilizados para justificar o uso de uma nova solução dela derivada.
- Pode ser utilizado de várias maneiras – O número de maneiras em que um sistema de RBC pode ser desenvolvido são quase ilimitadas. Sempre que casos antigos estiverem sendo reutilizados, RBC estará sendo utilizado.
- Pode ser aplicado a uma grande gama de domínios – Estando somente limitada a forma como os casos são representados, RBC pode ser utilizada em vários domínios de aplicação.
- Reflete o raciocínio humano – o paradigma utilizado pelo RBC é fácil de ser entendido por seus usuários por sua semelhança com o aprendizado humano.

As informações contidas em um sistema de RBC poderiam então ser fornecidas de acordo com um contexto mais específico para os projetistas durante a fase de projeto conceitual, e devido ao fator humano envolvido no processo de criação e seleção dos casos o entendimento das razões por trás das soluções apresentadas deverá ser mais intuitivo. A cada projeto realizado com a ajuda do sistema teria-se mais informação para os projetos posteriores, o que poderia ser potencializado através de um sistema com elementos colaborativos.

Existe uma discussão sobre o uso de RBC e seu efeito na criatividade das soluções geradas por estes sistemas. Por um lado, defende-se que o RBC espelha-se na cognição humana para a realização de soluções (AGRES; WIGGINS, 2015) sendo ferramenta que implementa a criatividade por meio de adaptação e junção de soluções anteriores, por outro, avalia-se que grande parte das aplicações concentra-se em corresponder soluções já criadas para novos problemas semelhantes, como argumentado por (ZHU et al., 2013).

Algumas outras pesquisas já utilizam o CBR em conjunto com outras técnicas de desenvolvimento de produtos para a análise de soluções criativas para o projeto, uma delas é a utilização do CBR em conjunto com o TRIZ, por exemplo o trabalho realizado por (ROBLES et al., 2010).

2.4 Computer Aided Innovation (CAI)

O uso de ferramentas no auxílio a criatividade não é novidade na área de planejamento de projetos, porém existe ainda a discussão na forma como tais ferramentas afetam o processo criativo. Para o estudo desta relação foi criado o campo da Inovação Auxiliada por Computadores (IAC).

IAC é o campo de pesquisa que tem liderado os esforços através das últimas décadas para desenvolver soluções computacionais para dar suporte às diferentes atividades no processo de inovação (LOPEZ FLORES et al., 2015).

O objetivo destas então é facilitar o uso da criatividade para a resolução de problemas de projeto.

O objetivo destas novas ferramentas CAI em desenvolvimento é auxiliar inovadores, inventores, designers, desenvolvedores de processos e gerentes em seu desempenho criativo, com a expectativa de mudanças nos paradigmas através do uso desta nova categoria de ferramentas de software (LEON, 2009).

A efetividade das ferramentas de criatividade na facilitação da geração de ideias é dependente de muitos fatores, tais como o domínio da aplicação, a complexidade e dificuldade do uso da ferramenta, as habilidades e treinamento que será exigido dos usuários e os potenciais riscos relacionados ao seu mau uso (YAN; CHILDS, 2015).

Ferramentas utilizadas especificamente para o projeto conceitual já compreendem as mais conhecidas ferramentas CAD:

No momento, a maioria dos usuários de CAD se contenta com o uso das mesmas ferramentas de design e interface para o projeto conceitual que usam para o projeto detalhado. Ao mesmo tempo, a crescente competitividade do setor comercial e a crescente complexidade dos sistemas estão criando uma maior pressão para soluções inovadoras (ROBERTSON; RADCLIFFE, 2009).

Alguns exemplos de projetos que tratam da fase conceitual, gerando ferramentas (ou frameworks) para o auxílio ao projeto foram encontrados: (LOPEZ FLORES et al., 2015) abre o caminho para um framework baseado na teoria TRIZ para gerar um framework denominado pelo autor de Open CAI 2.0. O mesmo ressalva a importância da colaboração na resolução dos problemas de projeto. (ZANNI-MERK; CAVALLUCCI; ROUSSELOT, 2009) Se dedicou a criar uma base ontológica para servir de suporte a ferramentas computacionais nos quesitos de aquisição de conhecimento e formulação de problemas (ambos vitais ao RBC). (TEIXEIRA; RÉGIO PIERRE DA SILVA; BRUNO, 2008) Elabora um sistema Web colaborativo para o Projeto conceitual de produtos.

No próximo capítulo serão descritos então os caminhos escolhidos para a pesquisa dentre os conceitos aqui descritos e as razões pelas quais estes foram selecionados.

3 Desenvolvimento

Neste capítulo será mostrado todo o desenvolvimento da solução proposta pela pesquisa e os passos necessários para alcançar o seu desenvolvimento.

3.1 Solução proposta

A pesquisa apresentada neste documento tem como objetivo desenvolver uma ferramenta de auxílio à fase conceitual do projeto de produto. Durante o levantamento da literatura relacionada ao problema, foi observada que seria de interesse da pesquisa a avaliação de métodos de Aprendizado de máquina e correlatos para uso no desenvolvimento da dita ferramenta devido à possibilidade de uso de dados já documentados de projeto no processo decisório.

O uso em específico do método proposto pelo RBC é motivado pela possibilidade de aproveitamento das soluções de projetos anteriores em novos problemas. Este aproveitamento no entanto deve seguir regras que não ofereçam restrições ao projeto e nem ao processo criativo, servindo como ferramenta e não como limitador da tarefa a ser cumprida.

A ideia da aplicação proposta nesta pesquisa é a avaliação e uso das informações de soluções dos projetos já inseridos na base de conhecimento da aplicação para a composição de novas soluções, porém de forma direcionada indicando ao usuário quais as soluções presentes nesta base que poderiam se encaixar no contexto do problema em questão.

A ferramenta proposta consiste de um software que consiga cadastrar casos manualmente para povoamento da base e também provenha funcionalidade de inserção de um problema e a opção de utilizar as funcionalidades de RBC para a proposição de uma solução na forma de uma matriz morfológica de soluções, com tanto o cadastro como a solução sendo apresentadas com o formato já conhecido pelos profissionais da área de Desenvolvimento de Produtos.

3.2 Requerimentos do software

Ao analisar o problema, percebe-se que a funcionalidade presente para o usuário no software deve ser simples, em seu mais alto nível. Como constatado por (YAN; CHILDS, 2015), citado anteriormente, as ferramentas devem ser de uso fácil e intuitivo evitando que dificuldades no uso interfiram com seu desempenho. Como decisão de projeto então foi estabelecido que o modo de uso deveria ser o mais próximo possível do método escolhido para o projeto conceitual, e também deveria ter seu uso simplificado e com funções de intercâmbio de casos para que os casos gerados durante os testes em ambiente acadêmico pudessem ser centralizados em uma só base. Destas direções principais foi gerada então a lista de requisitos mostrada abaixo:

- Cadastro dos casos
 - Cadastrar novos casos
 - Editar casos já inseridos
- Interface familiar aos usuários da área da Engenharia de Produção
 - Decomposição funcional que mostre de forma sucinta os desdobramentos de funções e permita rápida edição dos parâmetros das funções e seus desdobramentos
 - Matriz morfológica que permita a inserção dos conceitos:
 - Automaticamente, com a funcionalidade de sugestão
 - Manualmente, a partir de:
 - Conceitos previamente armazenados
 - Novos conceitos cadastrados imediatamente
- Prover informações sobre os casos cadastrados de forma rápida e simples
- Fornecer a solução de modo legível e permitir alterações prévias ao armazenamento de um novo caso

- Permitir a exportação da base para outros sistemas e importação de casos de outras bases (desejável)
- Arquitetura modular para permitir o uso das funções do motor em sistemas externos

Estes vão ser os requisitos utilizados na decisão da arquitetura de software. A etapa de testes mais adiante terá como parte da validação do desenvolvimento verificar se todos estes foram cumpridos de forma satisfatória.

3.3 Dependências externas

A ferramenta proposta, apesar de ter como base o RBC, tem que estar apta a realizar algumas operações utilizando aprendizado de máquina, especialmente na etapa de recuperação aonde terá que ser avaliado o vocabulário das funções para a atribuição de semelhança. Os frameworks mostrados Tabela 2 possuem tais características e são os mais recomendados (dentre os pesquisados) para o que é proposto neste trabalho:

Tabela 2 - Frameworks considerados para o desenvolvimento.

Framework	Linguagem	Autor	Descrição
TensorFlow	Phyton e C++	Google	Desenvolvida utilizando o conceito de <i>Data Flow Graphs</i> , tem grande flexibilidade.
Weka 3	Java	University of Waikato	Tem documentação extensa e uma grande comunidade.
Apache Mahout	Java	Apache Foundation	Já tem algum tempo desde o seu início (2011) e compreende vários algoritmos, podendo ser integrado em outras ferramentas como Scala, Spark, H2O e Flink.
DeepLearning4J	Java	SkyMind	Framework Open Source de aprendizado de máquina, bastante utilizado para desenvolvimento em Java

Vários destes frameworks possuem funcionalidades semelhantes, e a seleção ao final dos testes com cada um foi realizada em parte por conveniência. Os Frameworks que não possuíam implementação em linguagem Java foram descartados e dos restantes foi escolhido o *DeepLearning4J* por sua implementação do componente de análise aprendizado gramatical, Word2Vec (DEEPLARNING4J, 2016).

Para o controle e trabalho dos dados contidos na base, foi utilizado o framework Hibernate de persistência. O mesmo foi selecionado por prover com relativa facilidade uma abstração em código das funcionalidades da base, transformando facilmente as tabelas da base de dados em classes que podem ser utilizadas tanto para trafegar os dados via operações na base quanto em código diretamente (RED HAT; JBOSS DEVELOPER, 2016).

A linguagem da base de dados é escrita em SQLITE, uma variante mínima da linguagem SQL, projeto de código aberto, que é utilizada em projetos menores como aplicações *Desktop*, embarcados e *Mobile*. (HIPPI; DAN KENNEDY; MISTACHKIN, 2016). É uma linguagem que corresponde as necessidades do projeto, sendo de fácil implementação, amplamente documentada e possui bom desempenho.

3.4 Arquitetura de software

Para simplicidade foi adotada uma arquitetura MVC (Modelo, Visão e Controle) para o projeto, visto que a funcionalidade do software é simplificada para o usuário e sua verdadeira complexidade estará no cálculo das sugestões da matriz morfológica. Esta complexidade está transparente ao usuário, que deve somente inserir os dados do problema e avaliar as sugestões geradas pelo motor da aplicação.

O padrão MVC separa apresentação e interação dos dados do sistema, estruturando o mesmo em três componentes lógicos que interagem entre si. O componente de modelo gerencia os dados do sistema e as operações relacionadas a estes. O componente de visão define e gerencia como os dados

são apresentados ao usuário. Por fim o componente de controle gerencia as operações, repassando os comandos do usuário para os dois outros componentes. (SOMMERVILLE, 2010)

O modelo da arquitetura MVC é então ilustrado na Figura 11, de forma simplificada.

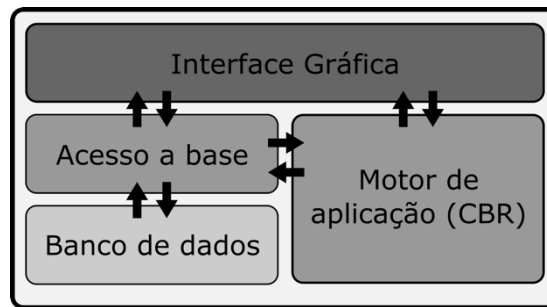


Figura 11 - Arquitetura de Software (simplificado)

Neste modelo de divisão da arquitetura temos, para o software produzido na pesquisa:

- Modelo – Forma como os dados são armazenados, compreendendo todos os *CRUDs* e as fórmulas de consulta aos dados e as transações de dados.
- Controle – são as ações efetuadas pela aplicação. Compreende também o motor da aplicação e os algoritmos do RBC.
- Visão – Abrange a interface gráfica / apresentação dos dados

As divisões de modelo e arquitetura foram elaboradas de modo a permitir portabilidade do motor de aplicação para outros sistemas, caso necessário.

Dentro desta divisão dos componentes, o projeto completo foi organizado em três projetos menores para facilitar o gerenciamento do código (CBRMecaControl, CBRMecaModel e CBRMecaView).

Além destes três projetos já mencionados, dois projetos auxiliares (fora do escopo de funcionamento do aplicativo final) foram criados para trabalhar os dados gramaticais utilizados para o treinamento do modelo gramatical.

Os três projetos principais se encontram disponíveis para a comunidade em um repositório no serviço GitHub (GITHUB INC., 2016), e o endereço do repositório no Anexo VI. Na Figura 12 é mostrada a página web com os nomes e divisão dos projetos no repositório:



Figura 12 - Repositório para compartilhamento dos códigos gerados no projeto

O ambiente de desenvolvimento foi montado para a codificação na linguagem de programação Java e para compilação foi utilizado o JDK versão 8 (ORACLE, 2011) sendo a IDE de escolha o Eclipse Luna, versão 4.4.2r2 (ECLIPSE FOUNDATION, 2016).

As bibliotecas e dependências necessárias para a compilação do projeto são gerenciadas utilizando o framework Maven (APACHE FOUNDATION, 2016) já integrado ao Eclipse e para todos os projetos foram criados os arquivos de configuração listando as dependências necessárias devidamente inseridas na raiz do projeto (no arquivo “pom.xml”).

Abaixo serão descritas as funcionalidades de cada um dos módulos e mostrados alguns detalhes da implementação para um entendimento mais amplo do desenvolvimento.

3.4.1 Modelo

A estrutura do caso mostrada na Figura 13 tem como elementos principais, que serão as entradas avaliadas para a solução do caso, as funções elementares.

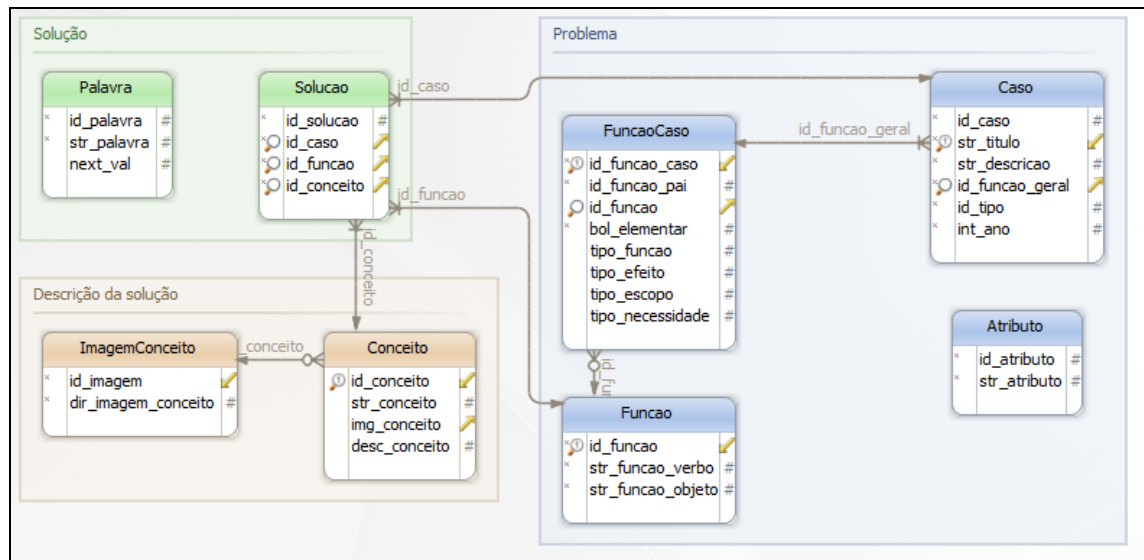


Figura 13 - modelo da base de dados para armazenamento dos casos

Um caso então tem uma estrutura principal que pode descrever um caso como um elemento com título, descrição, um conjunto de funções e para casos solucionados, também uma lista de conceitos associados as funções elementares.

Cada função possui o seu conjunto de parâmetros, que é utilizado de acordo com o mostrado em (AAMODT; PLAZA, 1994) da organização das características como um *feature vector*. Cada função na verdade é representada por dois itens em duas tabelas diferenciadas. Uma delas é a tabela de funções, que guarda somente a parte gramatical das mesmas, e outra a Função Caso, que guarda as particularidades da função utilizada para aquele caso em particular.

Alguns elementos adicionais foram inseridos visando a expansão das funcionalidades da ferramenta, como atributos para avaliação dos itens dos conceitos e que posteriormente pudessem ser usados para o desempate da decisão das soluções (manualmente). A tabela Atributo pode conter, por

exemplo, atributos como peso e tamanho relacionados com o conceito cadastrado.

Os conceitos podem ser cadastrados separadamente e acompanhados de uma imagem descritiva para prover no ambiente gráfico um auxílio visual, mantendo a consistência com as metodologias de projeto já consagradas entre os profissionais da área.

Uma tabela separada também foi criada para a inserção de vocábulos (Palavra), de forma a restringir a sintaxe utilizada em pontos-chave do processamento, como as descrições de funções. Estas, por vez, serão mantidas como um par verbo-substantivo para simplificar seu uso dentro do sistema.

3.4.1.1 Implementação

Para cada tabela da base, foram criadas classes que representam os dados nela contidos, que são chamadas de modelos (VOs), mostrados na Figura 14 e classes que gerenciam o seu acesso (DAOs), mostrados com sua hierarquia na Figura 15.

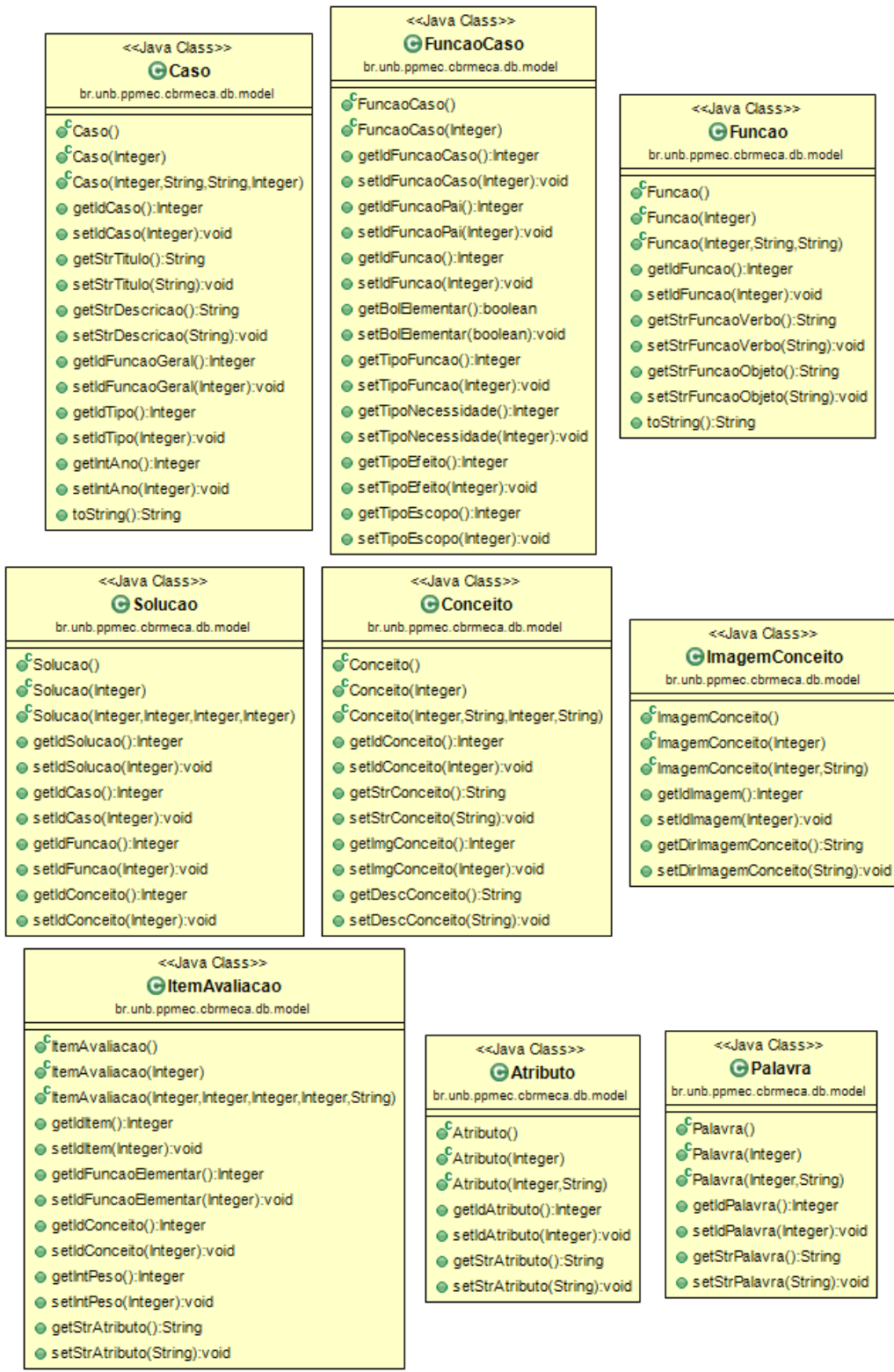


Figura 14 - Representação em classes da tabela do banco de dados (UML)

Todos os DAOs estendem de uma classe geral que provê o acesso de forma genérica, porém cada classe possui um DAO correspondente que implementa

pesquisas mais específicas que podem ser parametrizadas de acordo com as necessidades do programador para maior facilidade de uso.

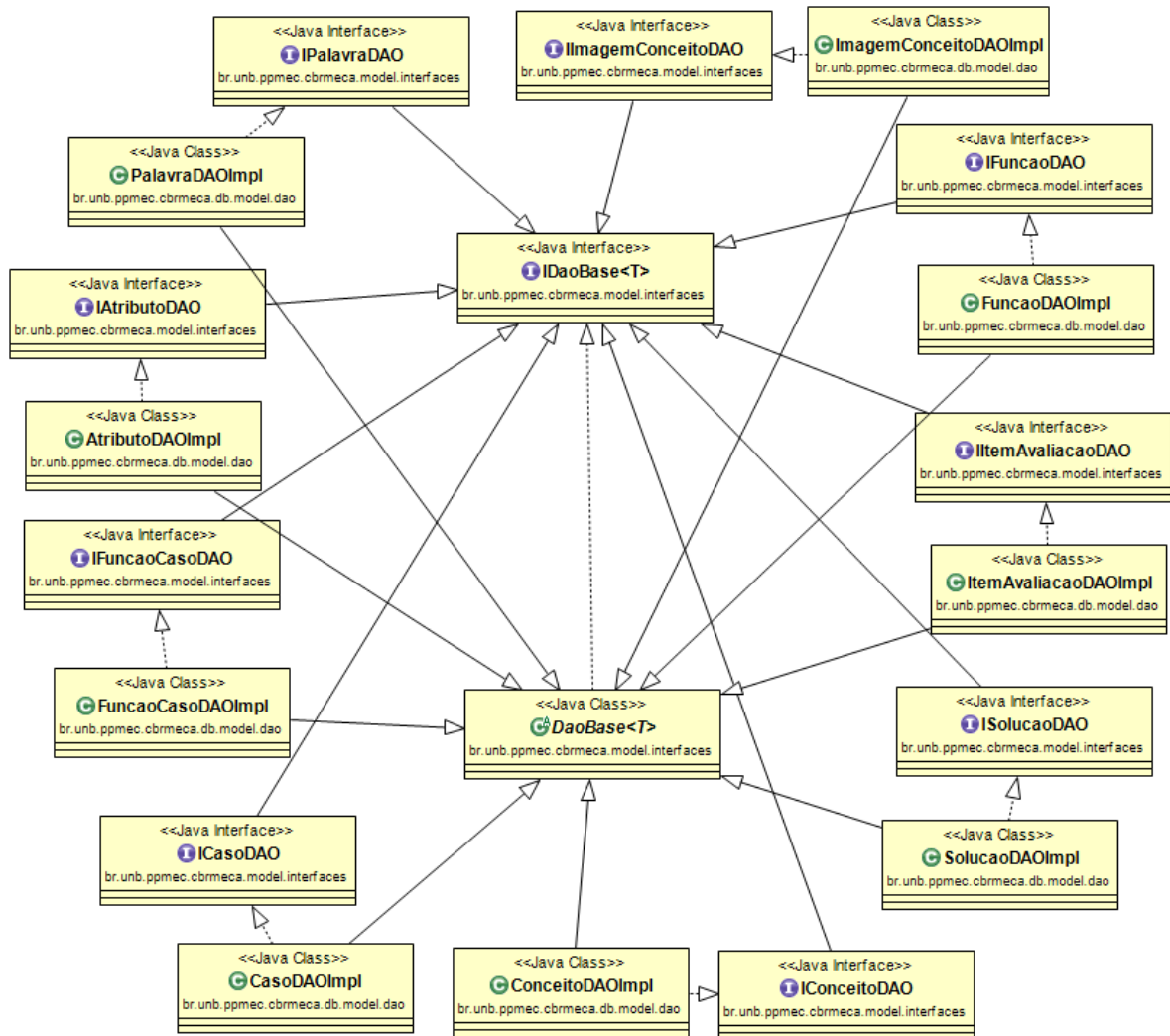


Figura 15 - Hierarquia dos DAOs (UML)

O acesso então é feito efetuando as chamadas para o DAO referente ao elemento da base a ser trabalhado e de acordo com o tipo de consulta implementado, por exemplo, uma consulta pode simplesmente retornar todos os elementos da tabela ou filtrado de acordo com a parametrização passada.

Algumas classes auxiliares foram criadas com os mesmos campos das classes que abstraem as tabelas para facilitar a conversão dos dados no padrão XML, necessários para persistir os dados em disco. As classes são mostradas na Figura 16. O modelo selecionado para a gravação será discutido mais adiante no documento.

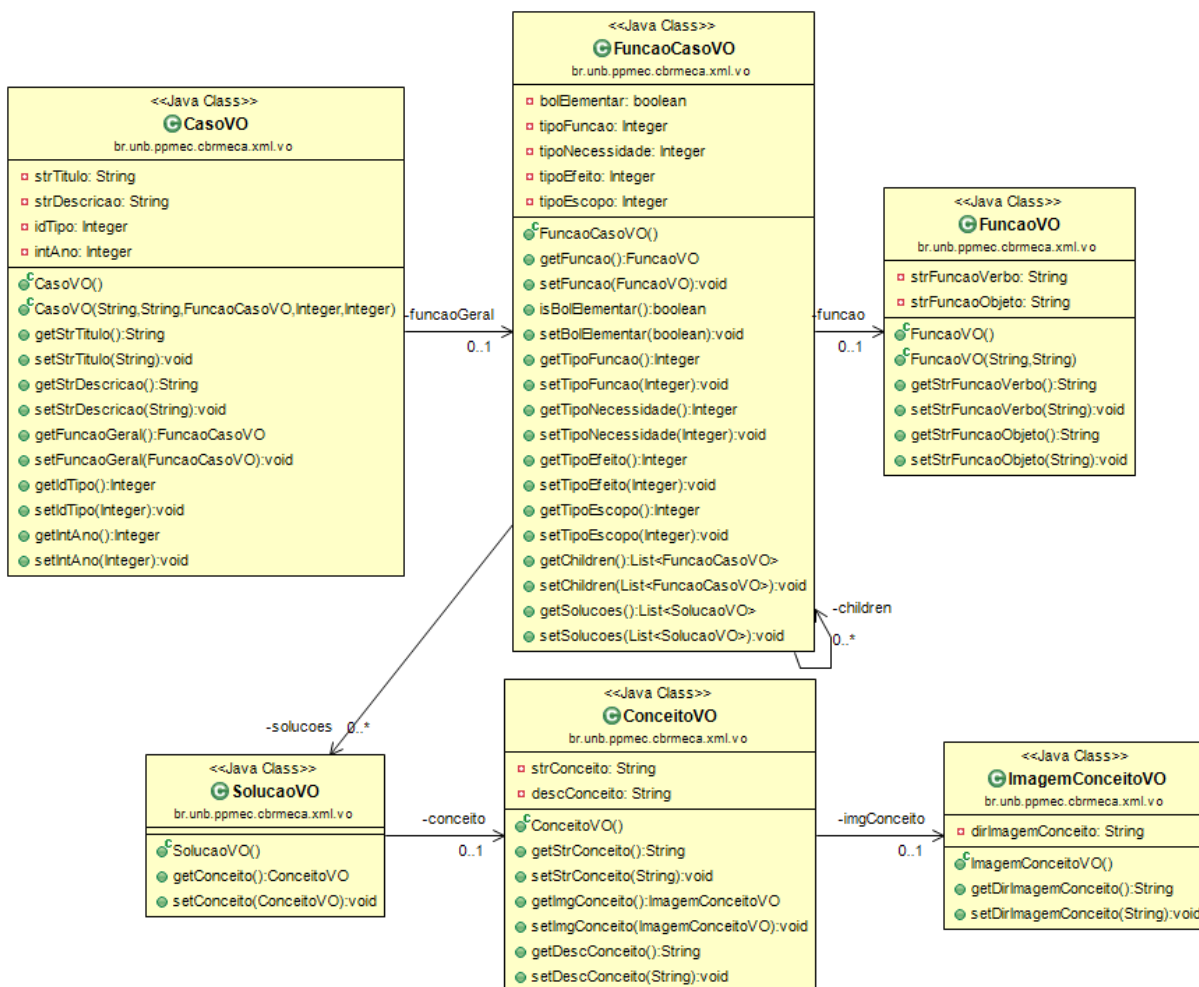


Figura 16 - Modelo de persistência dos casos (UML)

3.4.2 Controle

Este módulo é o mais importante para esta pesquisa, e compreende a ferramenta de sugestão em si. As tarefas realizadas neste módulo são as listadas no ciclo do RBC (LOPEZ DE MANTARAS et al., 2006), (AAMODT; PLAZA, 1994), e são elaboradas cada uma como métodos providos pelo motor. Abaixo são explicitadas as ações tomadas para o desenvolvimento de cada uma das etapas do RBC dentro da aplicação.

Recuperação

Existem várias opções de características para busca dos problemas semelhantes na base de dados, porém uma das principais dificuldades encontradas foi a necessidade de descrição dos casos com um conjunto fixo de

características. Avaliando a estrutura do projeto conceitual, a primeira opção lógica pareceu ser a análise das funções presentes nos casos para avaliação de semelhança, porém, as mesmas variam em relação a sua quantidade e raramente terão o mesmo número, o que levou a transpor o espaço do problema para as funções, estas sim passíveis de serem representadas por um número fixo de parâmetros.

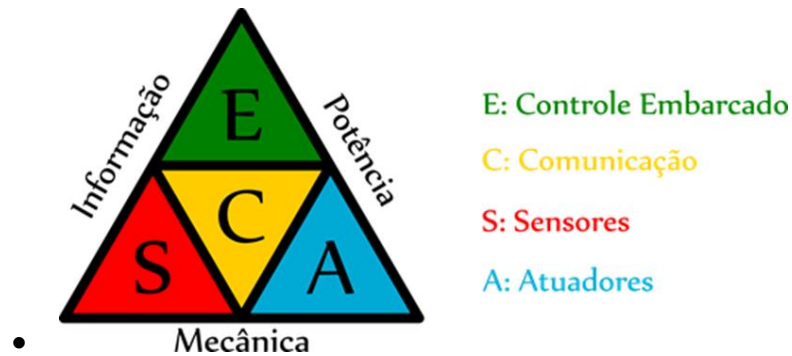
A maneira de representar as funções dentro do projeto então teve que ser avaliada, e dentre os fatores levados em consideração para sua classificação foram analisados quanto:

- Ao seu efeito dentro dos objetivos de projeto, servindo para amenizar ou corrigir efeitos oriundos da execução de outras funções (VEGTE; BREEMEN, 2009):
 - Regular – Não tem relação com efeitos indesejados de projeto
 - Preventiva – Previne efeitos indesejados
 - Corretiva – Corrige efeitos indesejados
- A sua necessidade, sendo avaliadas pela sua capacidade de servir aos propósitos do usuário (VEGTE; BREEMEN, 2009):
 - Necessária – Serve aos propósitos do usuário
 - Derivada – Criada somente devido as decisões de projeto
- O tipo de função de produto que esta representa, de acordo com a classificação apresentada em (ROZENFELD et al., 2006), ilustrada na Figura 17:



Figura 17 – Funções do produto adaptado de (ROZENFELD et al., 2006).

- Para que a função pudesse ser melhor avaliada no contexto de Sistemas Mecatrônicos, foi criado um parâmetro denominado “escopo” para avaliar e diferir qual o contexto disciplinar ao qual a função se adequaria. As disciplinas principais envolvidas neste contexto são ilustradas na Figura 18:



• Figura 18 - Áreas envolvidas na Engenharia Mecatrônica, adaptado de (ZHENG et al., 2014)

Algumas avaliações mais simplistas, denotam o sistema mecatrônico como uma junção entre Mecânica, Software e Eletrônica, como (ADAMSSON, 2005) na Figura 19:

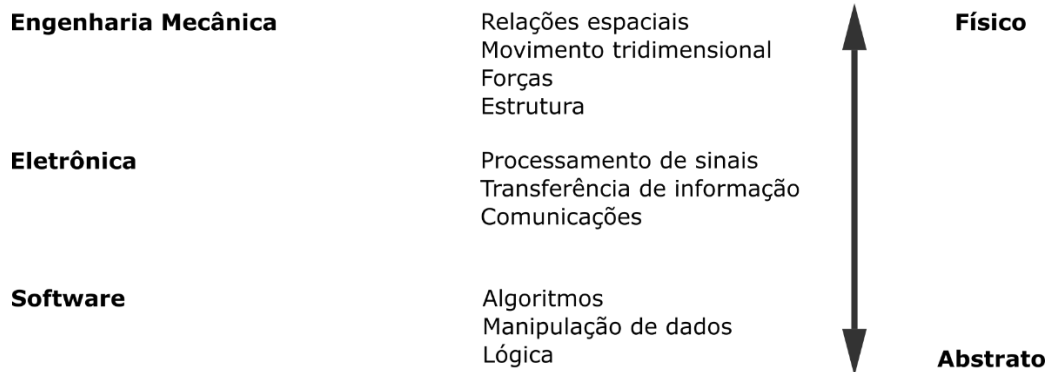


Figura 19 - Nível de abstração de tecnologias da Mecatrônica, adaptado de (ADAMSSON, 2005)

Usando o conceito do autor de separação por níveis de abstração, e fazendo uma pequena modificação, criamos então uma classificação semelhante, que denota como disciplinas de mais alto nível não só a mecânica, como também a elétrica como disciplina de alto nível, porém de modo que a elétrica englobe também a parte eletrônica.

O escopo então se define por cada uma das disciplinas envolvidas no projeto, com a adição de um item denominado “controle”, especificando

se o contexto possui ou não esta característica. É interessante notar que existem meios de se efetuar controle em qualquer uma das três disciplinas e não só através do Software (Controle Digital, Elétrico e Mecânico). A avaliação do escopo do contexto foi elaborada em termos das entradas e saídas (energia, material e sinal) e as grandezas das quais trata a função. A Tabela 3 demonstra essa divisão para elucidar o conceito:

Tabela 3 - Detalhamento do escopo disciplinar das funções

Denominação	Exemplos	Descrição
Informacional / software	Processamento e armazenamento de dados	Envolve transformação de informações em meio computacional, como lógica e algoritmos.
Mecânico	Aplicar força, amortecer impacto, Suportar estrutura.	Envolve a transformação de grandezas Mecânicas como força, massa e pressão, rotação e translação.
Controle	Ajustar direção, manter estabilidade.	Envolve o trabalho de sinal, através de controle. Aplicado em conjunto com as outras três.
Potência	Filtrar sinal, alimentar equipamento, modular frequência.	Envolve a transformação de grandezas elétricas como tensão, corrente e potência.

Juntamente a estas análises, é efetuada e com maior peso sobre a decisão da solução final uma análise gramatical das frases que compõem as funções, (guardadas na base na forma “verbo + objeto”) primeiramente avaliadas pelo verbo que as compõe, caso o verbo tenha funções com verbos similares na base, as mesmas passam a uma etapa posterior, aonde são buscados dentro do objeto novamente palavras com significados semelhantes aos que compõem o objeto da função alvo para recuperar um grau de semelhança.

A análise principal efetuada é a análise do vocabulário das funções, que apesar de ser básica foi selecionada por ter certa facilidade de desenvolvimento, com material extenso disponível e suporte nativo do framework selecionado para o trabalho. Para este trabalho foram gerados três modelos (também chamados de *Corpus*), cada um com aproximadamente cinco mil palavras treinadas a partir de um *dump* da *Wikipedia* (WIKIMEDIA, 2016) utilizando o formato Word2Vec. O modelo é intercambiável, podendo ser alterado para obter melhores resultados sempre que um mais extenso ou melhor adaptado for compilado.

O *dump* foi gerado utilizando os arquivos XML do backup de agosto/2016, dos quais foram extraídas as informações dos artigos separadamente e retirada toda a formatação e *tags* para extração do texto puro, as frases separadas e limpas foram gravadas em um arquivo de texto. Este arquivo de texto serviu de entrada para o framework e a partir dele foi gerado o modelo através da etapa de treinamento. A qualidade do modelo varia de acordo com o volume de dados de treinamento e a parametrização utilizada e pode-se ajustar, por exemplo, qual é a frequência mínima de um termo para que o mesmo possa ser inserido no *Corpus* (um termo teria que aparecer um determinado número de vezes na massa de dados de treinamento para ser contabilizado no treinamento). O modelo final é uma matriz na qual cada linha e cada coluna contêm os termos do *Corpus* e o seu conteúdo é a similaridade entre cada um destes. Estes valores são então acessados quando é solicitada a comparação entre dois vocábulos, retornando a similaridade associada a eles.

Em seguida, as funções que foram recuperadas da análise de vocabulário são comparadas para realizar uma melhor análise do seu contexto de uso, para a qual são utilizados os parâmetros definidos para a descrição das funções. O resultado destas comparações será um conjunto de funções consideradas semelhantes à função alvo da comparação.

Vale ressaltar que a caracterização dos dados por estes parâmetros inseridos no cadastro das funções tem como objetivo filtrar funções que sejam gramaticalmente semelhantes, mas que tenham contexto de uso

demasiadamente diferenciado, evitando ambiguidade causada pela comparação única dos vocábulos utilizados na primeira comparação. A similaridade entre as funções em relação ao contexto é feita utilizando um algoritmo que atribuindo a cada um destes um peso específico, relaciona então os fatores semelhantes que sejam iguais e relacionados a um peso e adiciona a uma pontuação que é somada e depois dividida pela soma dos pesos. É uma aproximação análoga ao algoritmo do Vizinho mais próximo (*Nearest Neighbor*), explicado por (MAIN; DILLON; SHIU, 2001) Na recuperação do vizinho mais próximo, o caso recuperado é escolhido quando a soma ponderada das suas características que correspondem a essa facilidade de consulta é maior do que os outros casos na base do caso. Em termos simples, um caso que corresponda ao caso de consulta em n número de recursos, será recuperado em vez de um caso que coincide com k número de recursos onde $k < n$. Alguns recursos que são considerados mais importantes em uma situação de solução de problemas podem ter sua importância denotada ponderando essas características mais pesadamente na correspondência.

Mais resumidamente, temos a relação apresentada por (FINNIE, 2002):

O algoritmo típico do vizinho mais próximo é mostrado na Equação 1:

$$\frac{\sum_{i=1}^n w_i \times \text{sim}(f_i^L, f_i^R)}{\sum_{i=1}^n w_i}$$

Equação 1 - Algoritmo do vizinho mais próximo

Reuso

Nesta etapa considera-se o fator humano na escolha das soluções a serem apresentadas. As funções listadas como semelhantes na etapa anterior do processo são mostradas ao usuário para que sejam selecionadas (caso o usuário considere necessário) somente as de maior semelhança. Depois de selecionadas, o sistema buscará na base todas as soluções de todas estas funções e as classifica de acordo com o maior número de utilizações. As soluções compostas serão então as que mais se adequaram ao uso dentro do contexto das funções apresentadas.

(AAMODT; PLAZA, 1994) O reuso de casos recuperados no mesmo contexto do novo caso se foca em dois aspectos (a) as diferenças entre o caso passado e o atual e (b) qual parte do caso pode ser utilizada para o novo caso.

O caso pode ser reutilizado através da cópia integral da solução, tentando aplica-la ao novo caso, ou ter uma etapa de adaptação aonde será feita análise e aplicação das partes cabíveis do caso recuperado ao novo. Ao solucionar as soluções manualmente, o usuário estará adicionando novas experiências na base que serão referência para os próximos projetos feitos utilizando o sistema, ou seja, adaptando a solução do caso recuperado ao problema vigente.

Revisão

Antes de definir quais serão as soluções finais, o usuário poderá revisar as soluções para avaliar se estas estão de acordo com as necessidades de projeto. Além do sistema de sugestão, também foi implementado um sistema simples de cadastro de conceitos que será usado para o cadastro de novos elementos quando o usuário não estiver satisfeito com as soluções criadas, alimentando a base com novas soluções.

Para melhor visualização, são guardadas uma imagem e uma descrição do conceito para a composição visual da matriz morfológica.

Retenção

A etapa final é a decisão da realimentação do caso na base de dados para uso futuro. O caso poderá ser guardado na base ou exportado para um arquivo padrão contendo os conceitos utilizados nesta solução e o projeto em si. O sistema idealmente deveria contar com uma base de dados remota aonde os casos seriam imediatamente armazenados a pedido do usuário.

Os algoritmos implementados para cada uma das fases do RBC podem ser intercambiados para melhorar a performance, através do uso da interface java que descreve as funcionalidades do ciclo. Idealmente teria-se vários algoritmos (especialmente para a fase que se mostrou mais pesada no caso deste projeto,

que foi a de recuperação) e que pudessem ser intercambiáveis desde que utilizassem o mesmo modelo de base de dados com as funções do módulo de modelo.

3.4.2.1 Implementação

O módulo de controle tem como responsabilidade prover a funcionalidade básica do aplicativo relativa ao RBC, provendo os cálculos de similaridade e retornando as soluções com base nas funções semelhantes analisadas.

A funcionalidade compreende duas funções básicas implementadas segundo a interface *ICBREngine.java*, provendo alguma flexibilidade, possibilitando implementações diferenciadas dos algoritmos de Similaridade e Recuperação de soluções. As classes que compreendem tais funcionalidades estão ilustradas na Figura 20.

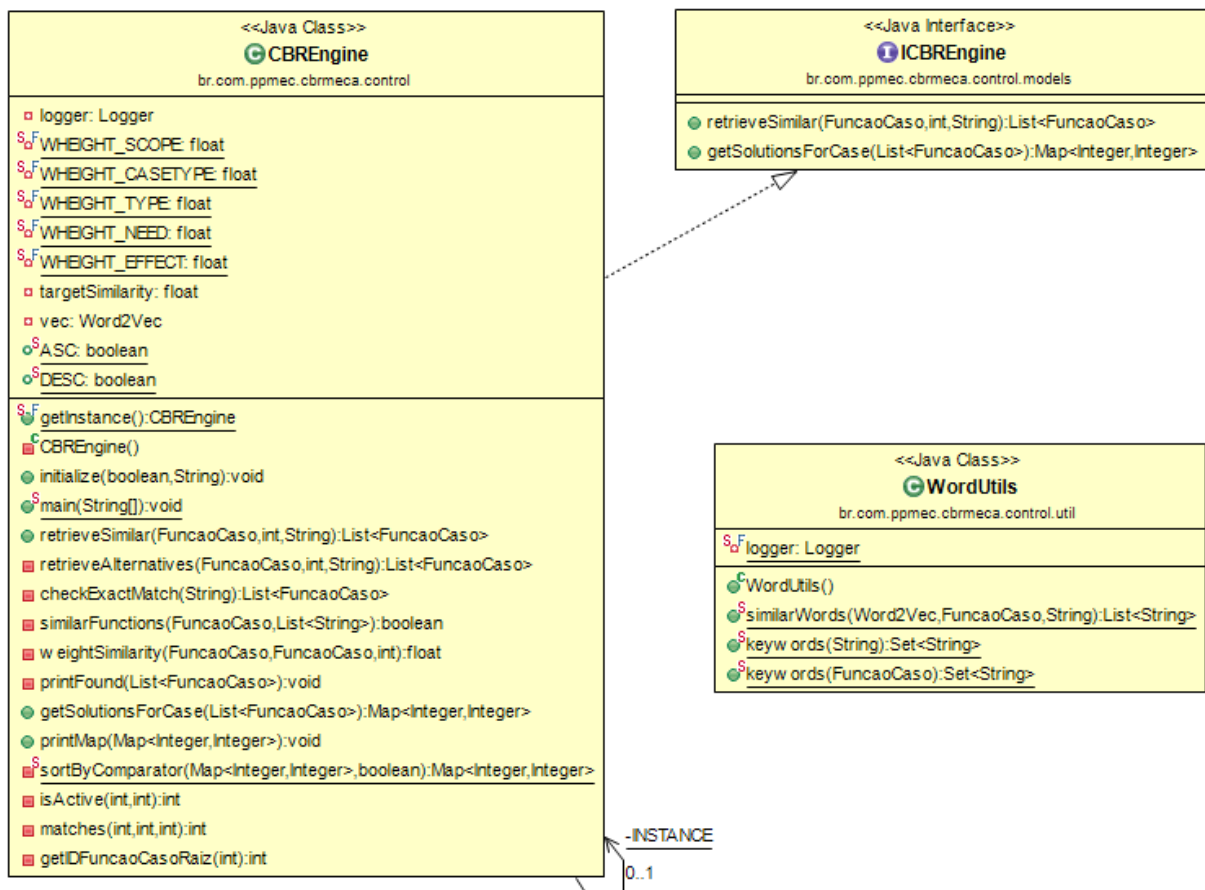


Figura 20 - Classes do projeto de controle (UML)

Esta particularidade tem como objetivo no futuro poder avaliar com mais facilidade outras implementações dos algoritmos de análise e sugestão, com a possibilidade de alterar os mesmos de acordo com a vontade do usuário e facilitar os testes e comprovação de desempenho dos mesmos.

As funcionalidades básicas do motor se limitam (externamente) aos métodos de recuperação de funções similares a uma função dada como entrada para retorno de uma listagem de todas as funções similares e outro método que lista, a partir de uma lista de funções todas as soluções compreendidas na base para estas em formato de uma lista única.

As implementações internas efetuam as operações referenciadas na seção anterior, seguindo o padrão inicial de busca por similaridade gramatical e posterior análise de similaridade dos parâmetros de classificação.

3.4.3 Visão

As funcionalidades do motor e da base de dados foram traduzidas na interface demonstrada a seguir com as devidas abstrações dos dados para visualização quando necessário. O trabalho do projeto conceitual requer que se tenha uma base gráfica para edição e modelamento das etapas de decomposição funcional e da matriz morfológica.

Para acompanhamento do crescimento da base, e sendo uma das primeiras etapas do desenvolvimento, logo no início somos apresentados a uma tela de estatísticas sobre os cadastros realizados e os elementos mais importantes presentes na base de dados. O que é mostrado nesta tela serve somente para acompanhamento e recuperação de informações para este projeto (Figura 21).

Ao usuário é dada a opção de buscar e rever os casos que já estão cadastrados para análise e exportação. Na versão atual do aplicativo é gerado um arquivo compactado com um XML descritor dos dados do caso, duas imagens (uma da árvore de funções e outra da matriz morfológica) e as imagens dos conceitos gerados pelo caso em questão a ser exportado. Desta forma, além de poder haver o intercâmbio dos casos, também serão

intercambiados os conceitos utilizados para a sua interpretação e para uso em casos futuros cadastrados em outras bases (Figura 22).

O modelo seguido é uma variação do apresentado por (ROZENFELD et al., 2006), e o processo foi simplificado em duas etapas, uma árvore de funções e a matriz morfológica de soluções (Figura 23).

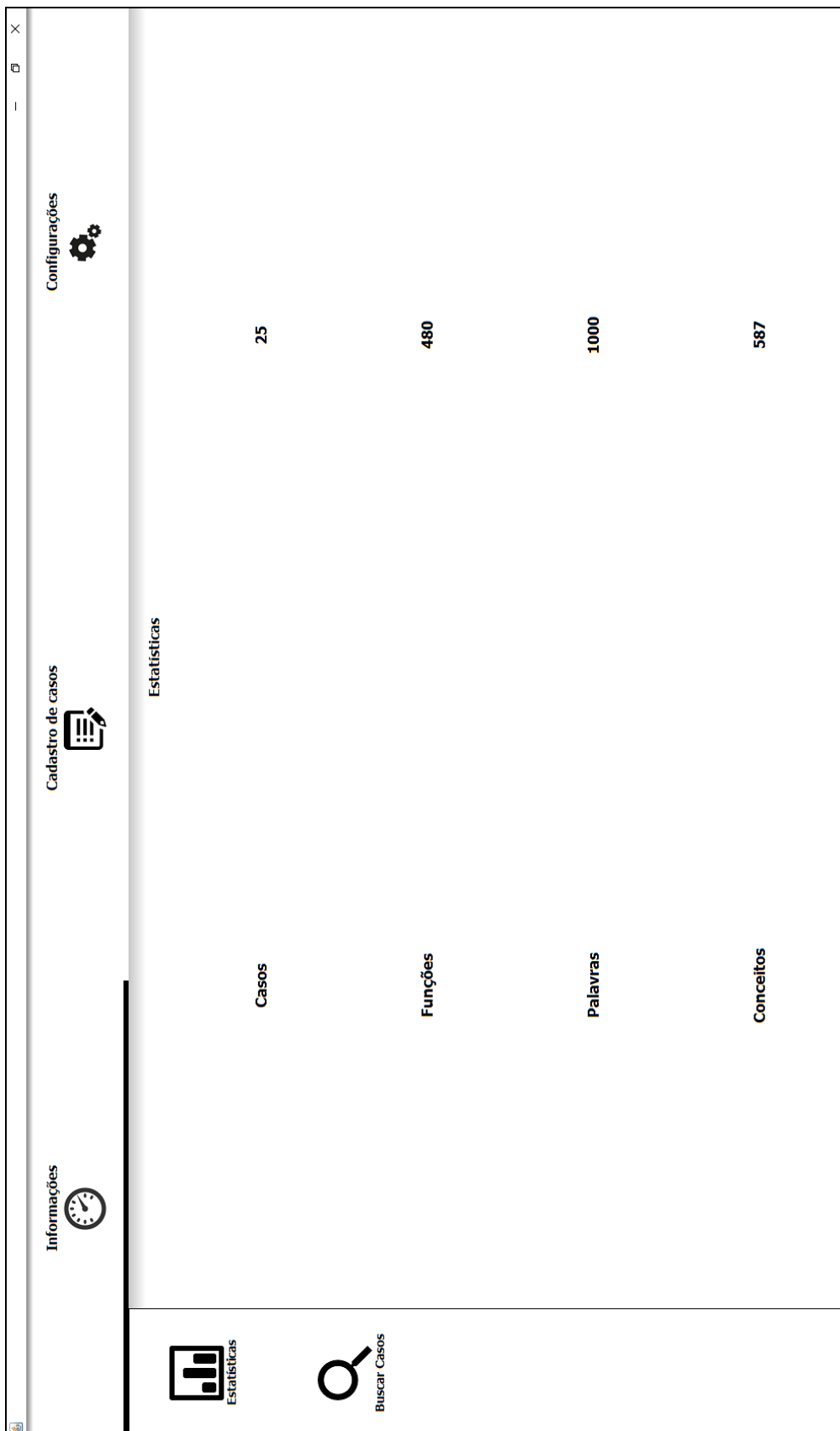


Figura 21 – Tela de estatísticas da base de dados

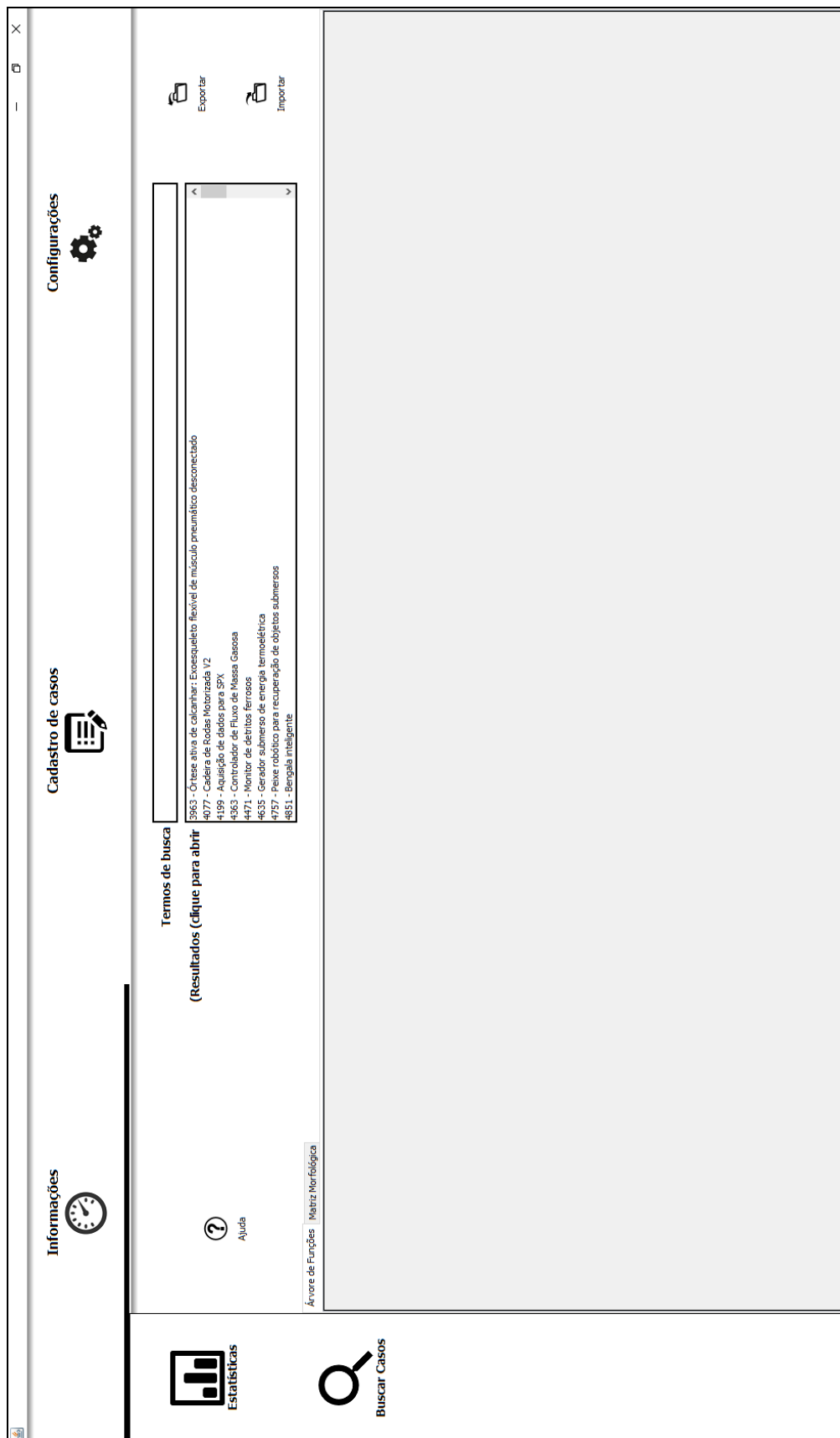


Figura 22 - Tela de busca e intercâmbio de casos.

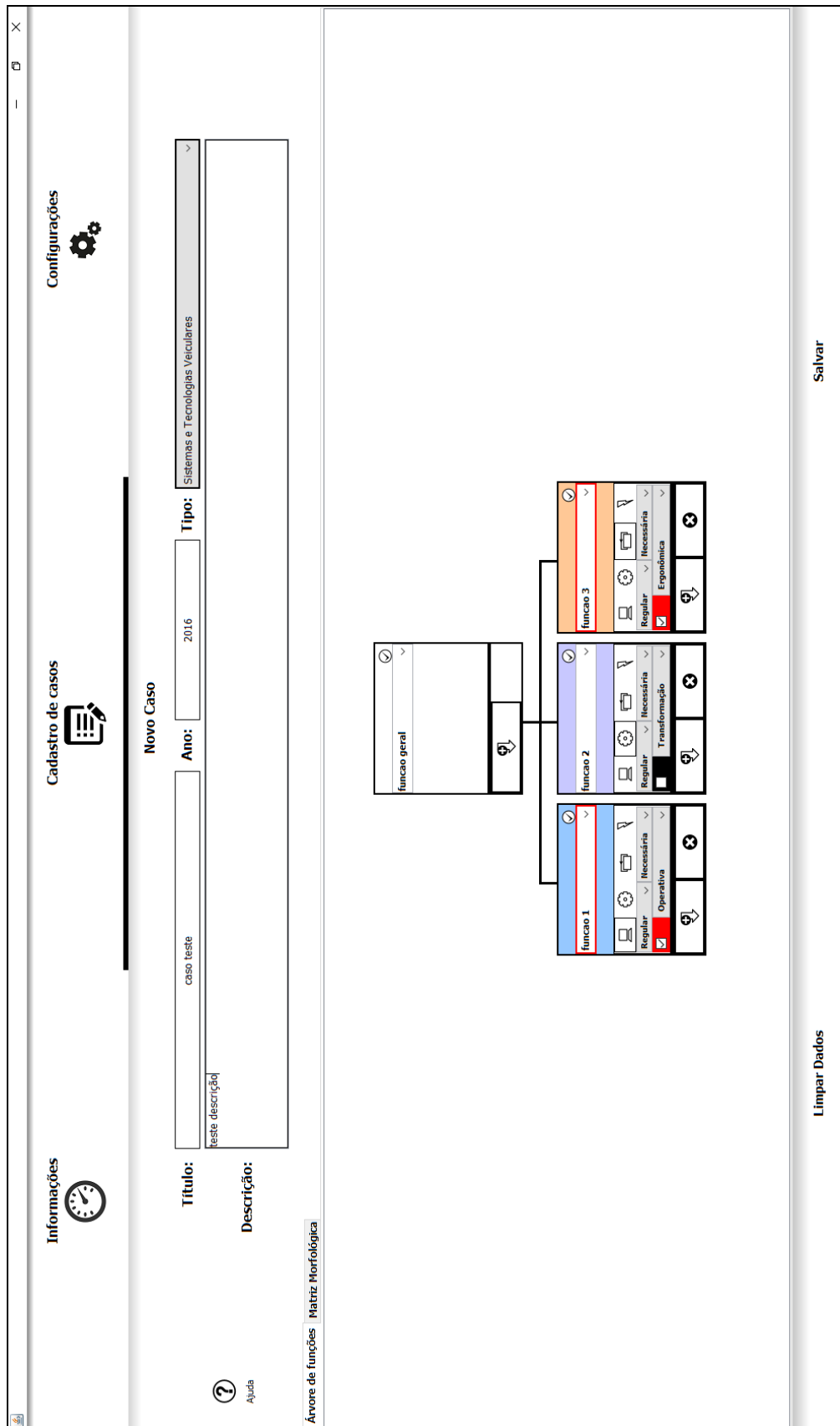


Figura 23 - Tela de cadastro de casos

O usuário deve primeiro preencher a árvore de funções com as informações de classificação e seus relativos desdobramentos. A função geral não possui classificações, e terá somente os seus desdobramentos sendo necessário que a árvore de funções tenha pelo menos dois níveis de desdobramento para que possa ser considerada válida (Figura 23).

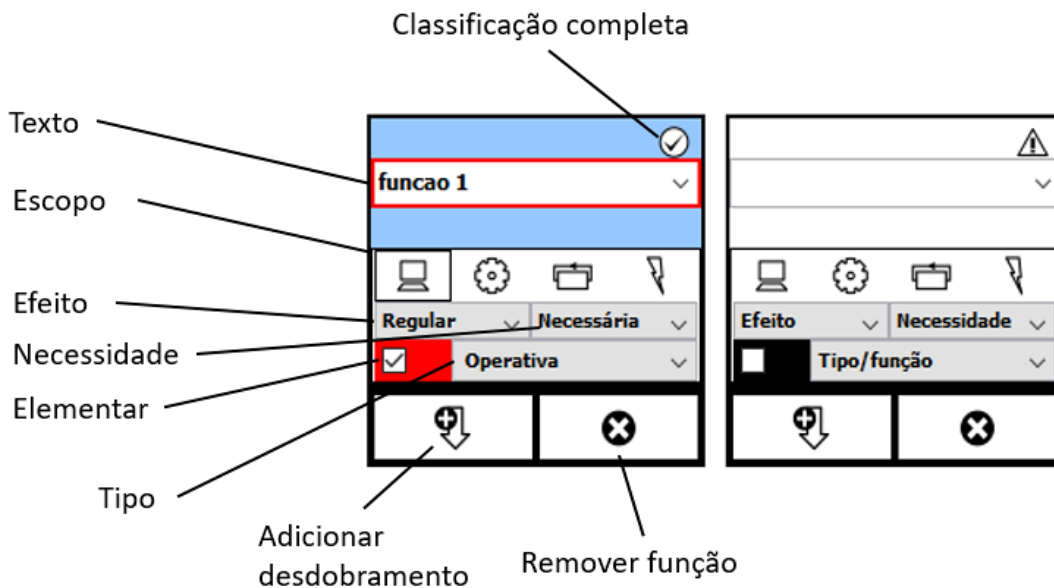





Figura 24 - Caixas de função completa e incompleta

Cada uma das funções adicionada á arvore deve ser preenchida de forma a ter o seu estado alterado, o que é mostrado visualmente no formulário. Após o seu preenchimento, a função deve ser finalmente classificada como elementar ou não. Caso seja marcada na caixa de seleção como elementar, ela será guardada para uso na matriz morfológica na próxima etapa do processo (Figura 24).

A matriz morfológica é a última etapa do processo e consiste na seleção, para cada uma das funções elementares, das soluções que irão compor a matriz. São oferecidas duas possibilidades para a seleção dos conceitos, uma é a busca e utilização manual de um dos conceitos na base (com a possibilidade do cadastro de outros conceitos na hora da inserção), e outra que é o foco principal do trabalho que é a sugestão de conceitos para “solucionar” a função.

Informações 

Cadastro de casos 

Configurações 

Novo Caso





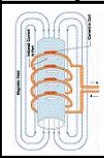






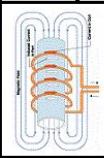


Título:

Ano:

Tipo:

Descrição:

Anore de funções: **Matriz Morfológica**

funcao 1	 Camera	 Laser azul	 Energia Solar	 Bateria Niquel-Cádmio	 Indução RF	 Sugestões  Adicionar Conceito
funcao 3	 Bateria removível	 Bateria recarregável	 Bateria LiPo	 Bateria Niquel-Cádmio	 Indução RF	 Sugestões  Adicionar Conceito

Limpar Dados

Salvar

Figura 25 - Editor da Matriz morfológica

São mostradas ao usuário as funções que o algoritmo julga semelhantes (Recuperação) para que o mesmo possa ter parte no processo de seleção e dentre destas aceitar todas, as que julgar mais apropriadas ao projeto, ou simplesmente procurar por si mesmo as que melhor se encaixariam ao projeto, mantendo assim o elemento humano participante no processo (Figura 26).

A interface da matriz morfológica também tem como base o modelo apresentado em (ROZENFELD et al., 2006).

As funções selecionadas serão utilizadas para recuperar as suas soluções e compor a solução final para a função da matriz. A solução sugerida pode ser editada, adicionando ou removendo conceitos de acordo com a necessidade do usuário no editor da matriz morfológica (Figura 25).

Ao final do processo, o usuário poderá salvar o projeto no arquivo de formato padrão citado anteriormente e também selecionar se o caso será salvo na base de dados para aproveitamento futuro, compondo assim a etapa final do RBC de retenção.

O fluxo para o uso do aplicativo é então mostrado abaixo:

- Inserir os dados iniciais do caso na tela de cadastro
- Inserir as informações da função geral
- Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
- Alterar para a aba da matriz morfológica
- Preencher a fila de conceitos para cada uma das funções utilizando as opções:

- Sugestões:
 - Selecionar as funções para uso na recuperação das soluções
- Adição de conceitos
 - Selecionar, utilizando a busca um dos conceitos já presentes na base
 - Adicionar um novo conceito
 - Inserir uma descrição para o conceito
 - Inserir uma imagem para o conceito (opcional)
- Salvar o caso, escolhendo um local para gravar o arquivo com sua descrição.

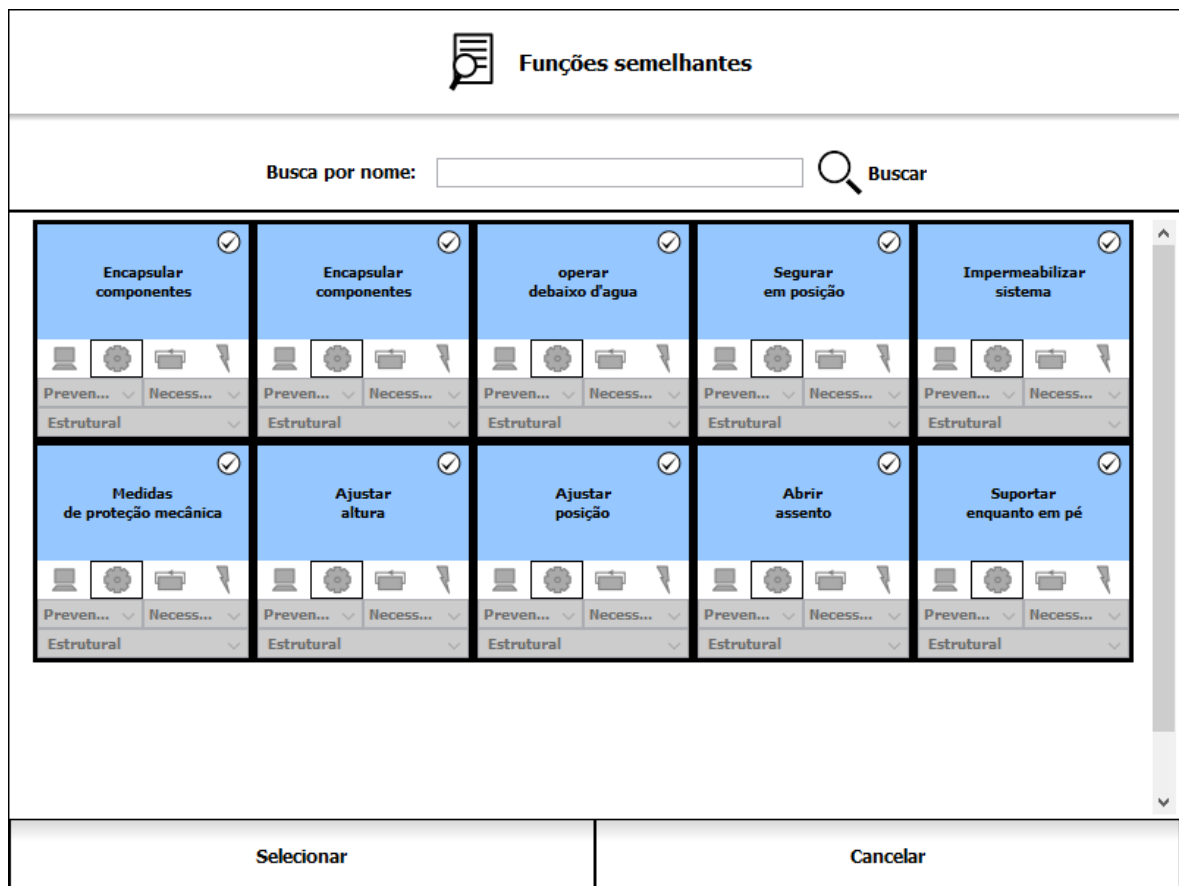


Figura 26 - funções semelhantes para seleção de soluções

A funcionalidade de sugestão teve a maior preocupação durante o desenvolvimento da ferramenta, pois dela advém a etapa automatizada do processo de recuperação da base das funções desejadas. Esta é a funcionalidade que compõe o processo de Recuperação do RBC, como descrito pelos autores citados na revisão da literatura.

O resultado final do desenvolvimento para esta funcionalidade pode ser descrito através dos exemplos a seguir. No primeiro, temos uma função bastante comum (utilizada em vários dos projetos) e que ilustra o funcionamento esperado idealmente para a sugestão de funções (Figura 27). Esta função, por ter um conjunto mais variado de casos utilizados tem uma grande quantidade de conceitos para utilização, bem como uma grande quantidade de variações, ou seja, possui um número mais variado de contextos de uso da mesma denominação “Alimentar dispositivo”, aumentando as chances de que soluções válidas sejam apresentadas.

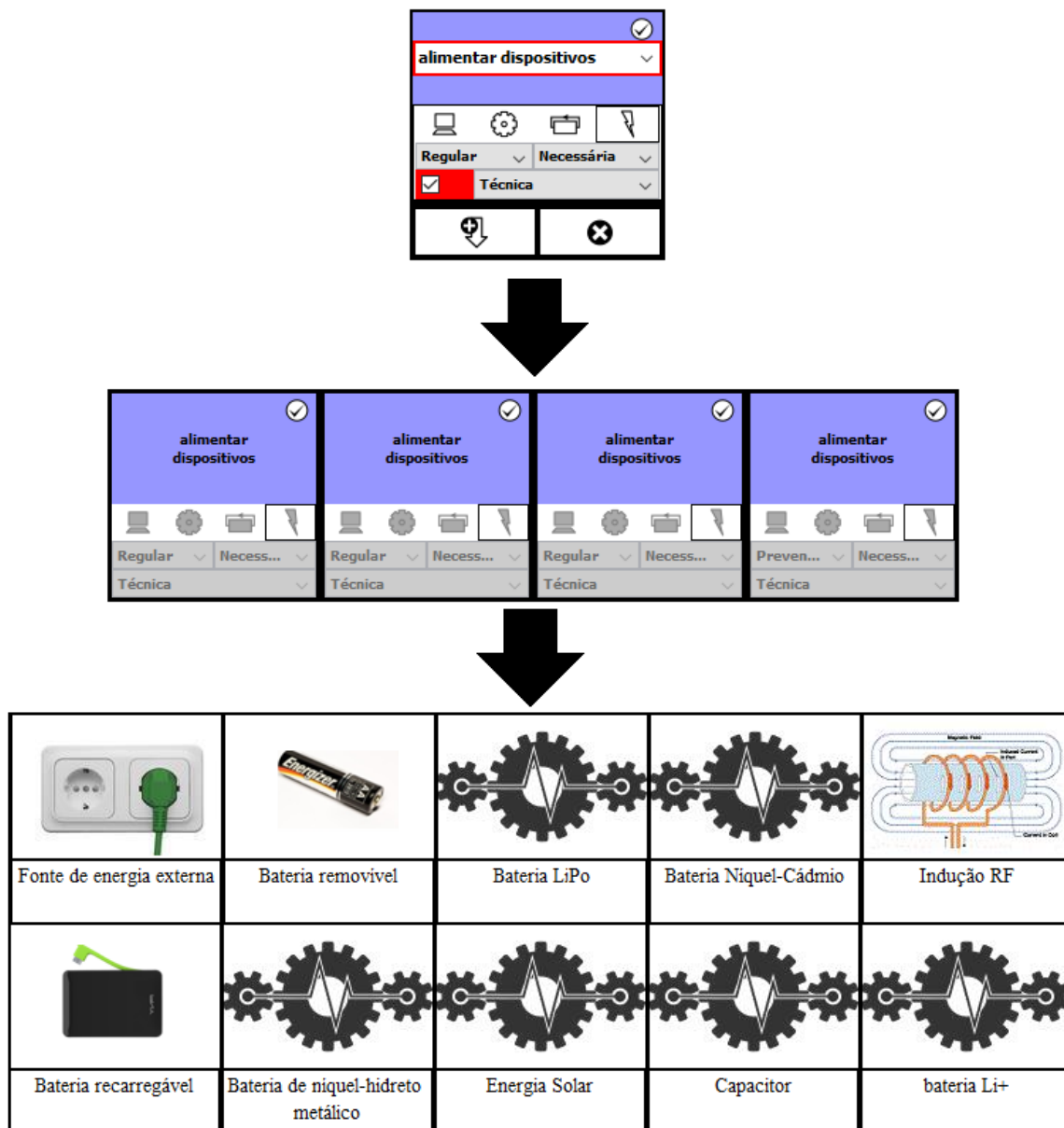


Figura 27 – Sugestão automática dos conceitos para a função "Alimentar dispositivo"

A alteração dos parâmetros da função neste caso ilustra claramente a diferenciação entre as sugestões para cada uma das situações, como mostrado na Figura 28:

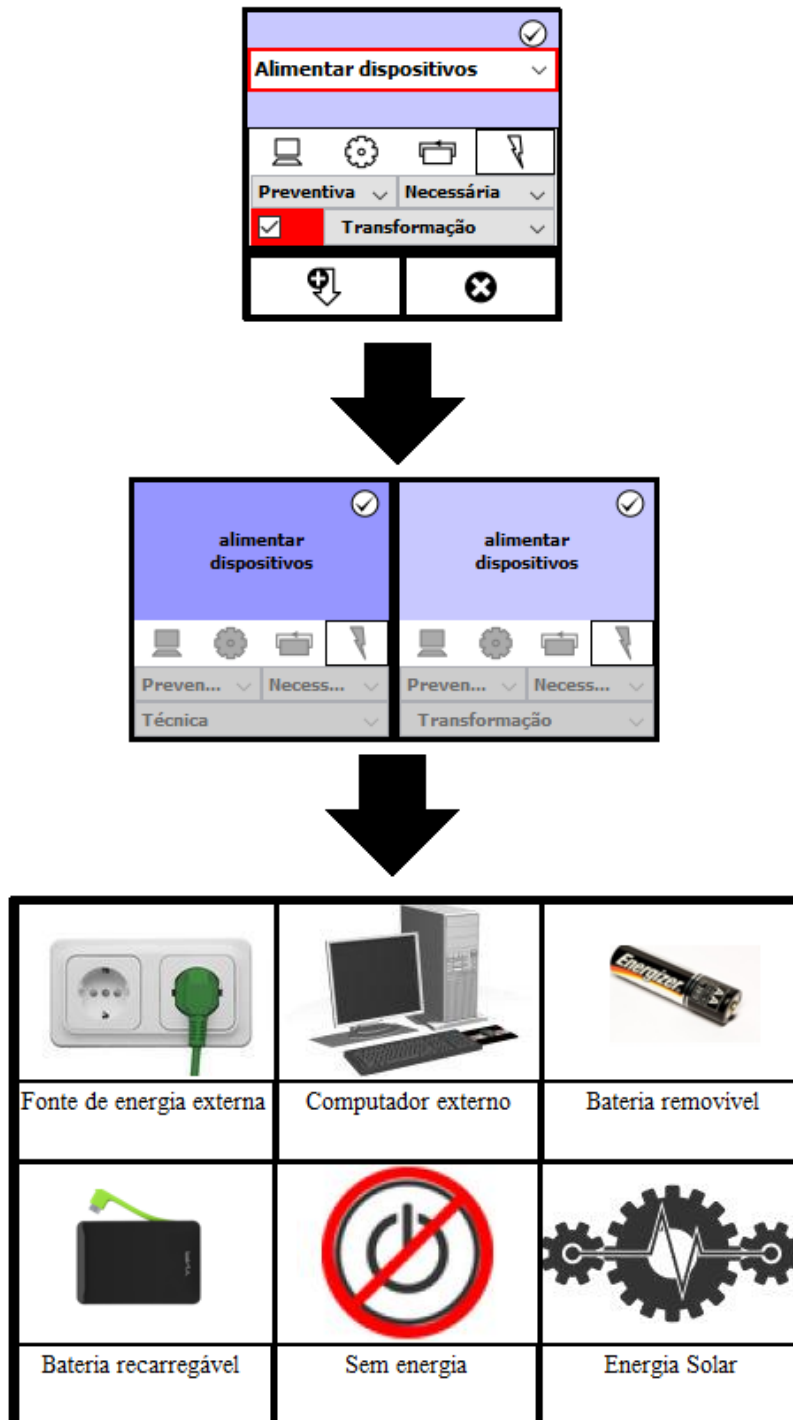


Figura 28 - Parametrização alternativa para a função "Alimentar dispositivo"

Ao sugerir os conceitos, a ferramenta é capaz de analisar então tanto gramaticalmente quanto contextualmente as condições em que as funções foram utilizadas e propor já com alguma precisão conceitos relevantes para o projeto. Para as funcionalidades acima analisadas vale ressaltar que a composição gramatical da função pode variar, e desde que mantendo o

significado a ferramenta ainda conseguirá entender como uma função semelhante. Um outro exemplo é a função sintaticamente semelhante “Fornecer energia”, que nos mostra os resultados familiares ilustrados na Figura 29:

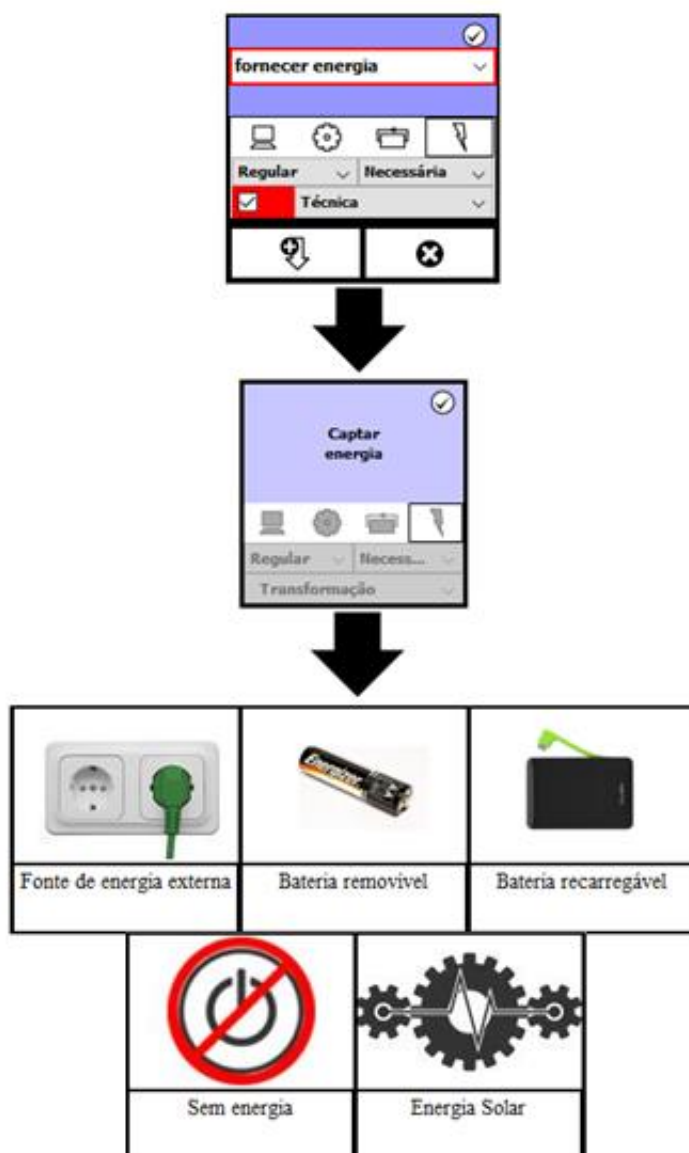


Figura 29 - Funções semelhantes a "Fornecer energia"

Os conceitos resultantes terão maior variedade de acordo com as características e quantidade dos projetos relacionados na base. Para as funções ilustradas acima os conceitos sugestionados foram bastante semelhantes, mas deve-se isto à solução semelhante das duas funções. O sistema pode por exemplo compor soluções com mais de uma função, desde

que ela tenha cadastrada funções gramaticalmente e contextualmente semelhantes. Mais um exemplo é dado abaixo mas desta vez para a função “Proteger aparelho”, ilustrada na Figura 30:

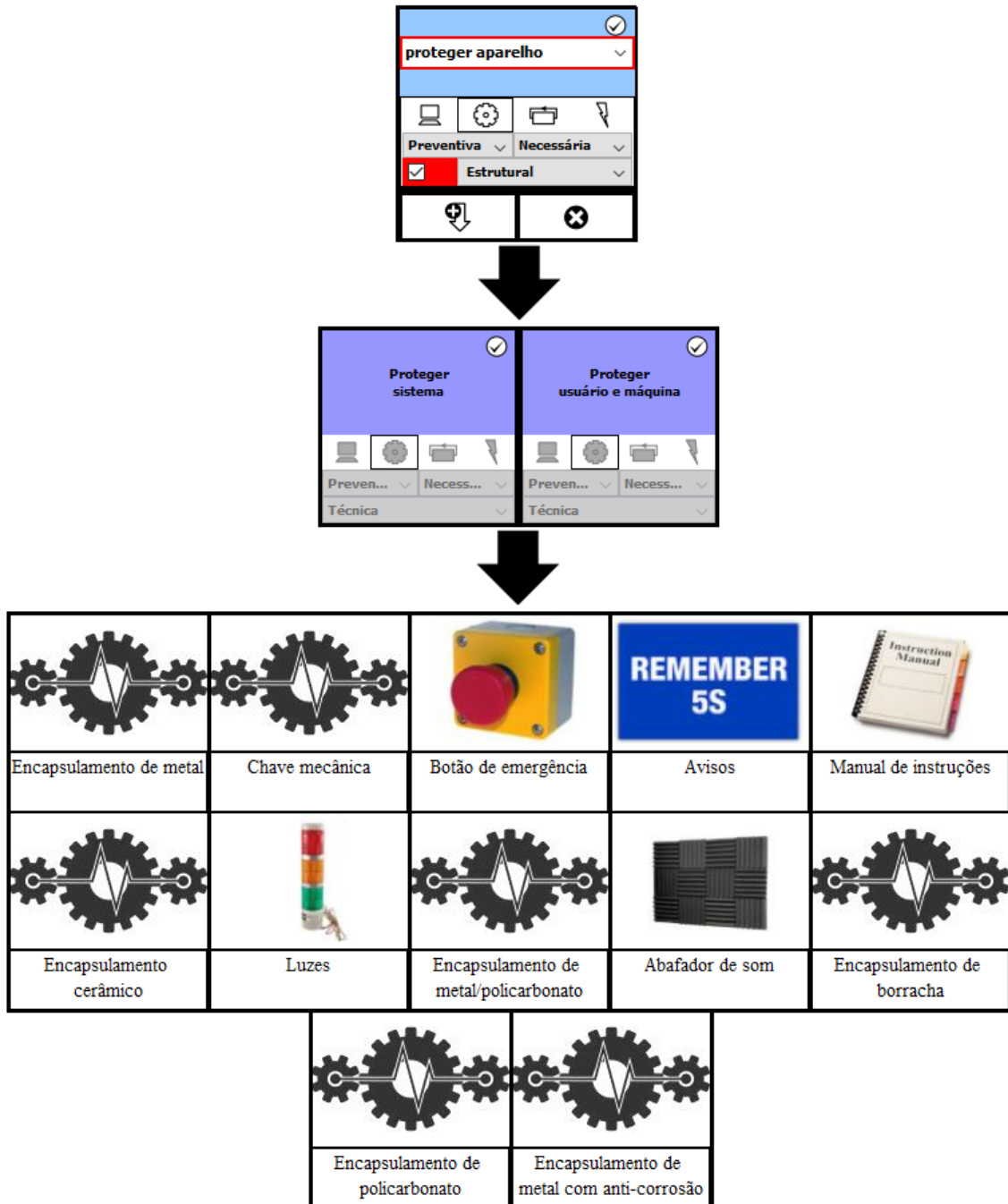


Figura 30 - Exemplos para a função "Proteger aparelho"

Obviamente a quantidade de conceitos retornada será proporcional à quantidade de funções utilizada para a geração dos mesmos. O usuário ainda assim não está preso aos conceitos trazidos e os mesmos poderão servir

somente como referência para a solução. O usuário poderá simplesmente descartar todos e gerar uma solução própria, se assim desejar. A variedade é um elemento importante para o sistema e deixando a encargo do usuário usar ou não as sugestões poderá servir para melhorar o aprendizado da base segundo a metodologia do RBC e melhorar a criatividade da solução gerada.

Ao final da implementação, foram cumpridas os principais requisitos listados no início deste capítulo. A base de dados inicial para o uso no processo de validação contém já informações de casos cadastrados retirados de um servidor de projetos do Instituto de Tecnologia de Rochester (RIT, [s.d.]) e de Projetos dos Alunos da Professora e Orientadora deste projeto, Andrea Santos.

Todos os projetos tem base mecatrônica e foram selecionados de acordo com sua qualidade, quantidade de informação contida na documentação (alguns casos observados nas fontes citadas não se encontravam completos) e diversidade de aplicações:

- Aparato de testes para próteses de mão
- Aparelho para detecção de crise asmática em crianças de 1 a 3 anos
- Apontador Laser Estabilizado ativamente
- Aquisição de dados para SPX
- Bengala inteligente
- Cadeira assistiva adaptável ao terreno
- Cadeira de Rodas Motorizada V2
- Capacete/software para encefalograma compatível com MRI
- Carregador de veículo elétrico off-board
- Controlador de Fluxo de Massa Gasosa
- Controlador de motor trifásico
- Copo com sensor
- Gerador submerso de energia termoelétrica

- Hélice de propulsão magnética
- Interface ambiental ADC
- Monitor de detritos ferrosos
- Máquina de medição da qualidade de cafés especiais
- Peixe robótico para recuperação de objetos submersos
- Processador de áudio híbrido analógico/digital
- Roteador de PCs
- Simulador robótico de movimento ocular
- Sistema de testes dinâmico para rolamentos de turbina
- Sistema universal de aquisição de dados de Bombas de fluido
- Varal automatizado
- Órtese ativa de calcanhar: Exoesqueleto flexível de músculo pneumático desconectado

3.4.3.1 Implementação

A estrutura do módulo é composta das classes mostradas na Figura 31. Alguns componentes específicos foram criados para suprir algumas funcionalidades de forma a facilitar os testes da interface antes de sua integração ao aplicativo. Por exemplo, o componente *JpanelFunctionClassification* contém o painel de classificação das funções, bem como métodos acessórios para a recuperação dos itens marcados pelo usuário. De forma semelhante temos o componente *CaseMatrixElement*, que é a representação do conceito inserido na matriz de soluções. Ambos estes componentes são reutilizados na tela de busca dos casos, representada pela classe *CBRDashSearchCasesPanel* e na tela de cadastro dos casos, representada pela classe *CBRMecaCasesPanel*. A diferenciação entre ambos é a seleção ou não da opção de edição e os eventos envolvidos na ação do componente correspondente.

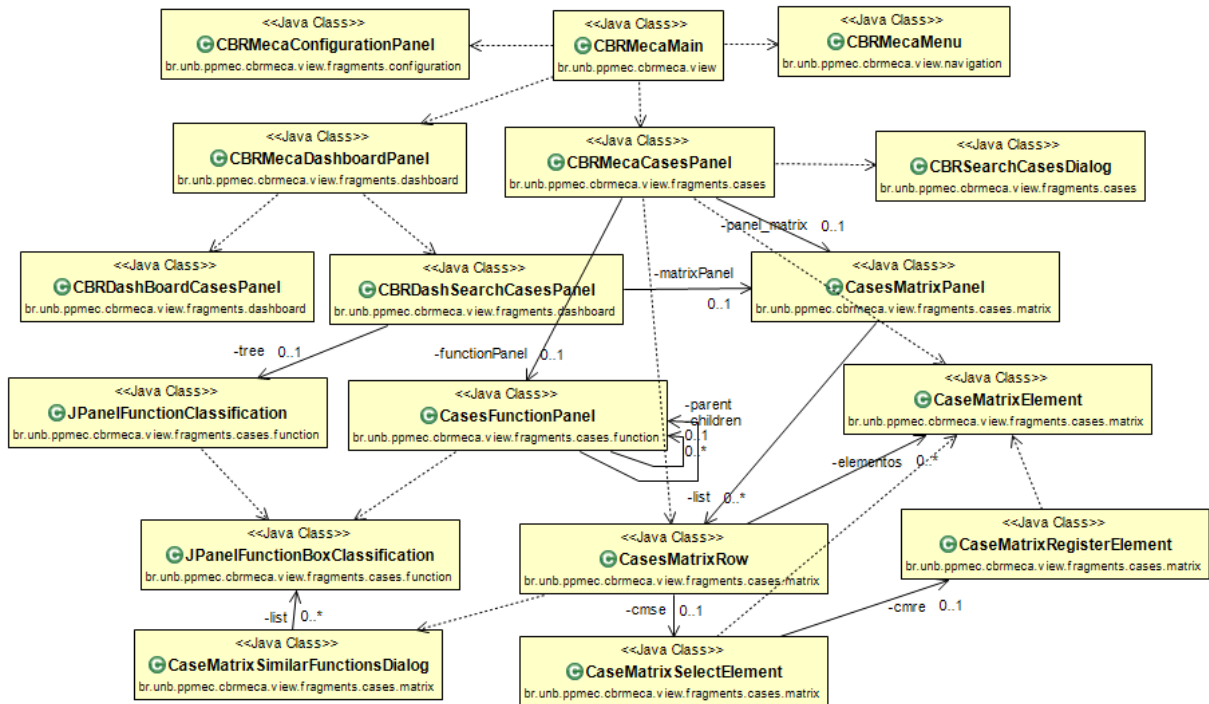


Figura 31 - Estrutura dos componentes do módulo de Visão(UML)

A comunicação entre os componentes utiliza em alguns casos eventos para informar as ações efetuadas pelo usuário e efetuar as chamadas para os outros módulos.

A parte final do desenvolvimento será a avaliação do software. No próximo capítulo será mostrado o modelo criado para a validação do software juntamente com a análise dos dados gerados dos testes.

4 Resultados

Neste capítulo será mostrada passo a passo a validação da ferramenta desde o planejamento até execução final.

4.1 Metodologia de avaliação

O objetivo desta etapa é avaliar principalmente se os objetivos do projeto foram alcançados. Parte deste processo envolverá avaliar tanto o desempenho do software em si quanto o seu impacto na etapa de projeto proposta em relação à facilidade com que sua execução foi efetuada e a qualidade da solução alcançada.

(FDA, 2002) trabalha os conceitos de Validação e Verificação como duas etapas separadas sendo cada uma delas com propósitos específicos e em acordo com a norma ISO 8402:1994.

A validação do software é a confirmação por exame e provisão de evidência objetiva de que as especificações de software estão de acordo com as necessidades do usuário e as utilizações pretendidas e que os requisitos específicos implementados por meio do software podem ser consistentemente cumpridos (FDA, 2002).

A verificação de software avalia a consistência, integridade e corretude do software e sua documentação, conforme os avanços no desenvolvimento, e fornece suporte para uma conclusão subsequente de que o software está validado (FDA, 2002).

Os objetivos de cada etapa para esta pesquisa serão, baseados nas prerrogativas acima:

- Validação – Avaliação focada nas necessidades do usuário e se elas foram atendidas com o uso da ferramenta. Envolve a opinião do usuário em relação a sua experiência durante o experimento.

- Verificação – consiste em avaliar se os requisitos iniciais de projeto foram atendidos. A avaliação envolve o teste de cada uma das funcionalidades relacionadas ao requisito ou requisitos correspondentes.

(SOMMERVILLE, 2010) generaliza os conceitos de Verificação e validação de software, dividindo as etapas de testes em 3 outras distintas:

- Testes de componentes – são testes realizados durante o desenvolvimento em componentes de software separado para verificação de seu comportamento individual.
- Teste do Sistema – é o teste do sistema já integrado, ainda em desenvolvimento.
- Teste de aceitação – São testes feitos fora do ambiente de desenvolvimento, também chamado de teste alfa.

O planejamento de testes desta ferramenta consistiu em unir as etapas de testes de componentes e do sistema em uma única etapa, a verificação, e o teste de aceitação como a etapa de validação.

Nos próximos itens serão explicados em maiores detalhes do que compreenderá cada etapa.

4.1.1 Validação

A Validação terá como foco principal o ambiente de sala de aula. Para tanto quatro grupos da disciplina de Graduação de Projeto de Produto de Sistemas de Produção 6 (Disciplina de projeto de produto) foram instruídos no uso do sistema para que pudessem efetuar o projeto conceitual do trabalho do semestre utilizando a ferramenta. No momento do uso todos cursam a disciplina de PSP6, que tem como parte do conteúdo letivo o Projeto conceitual de produtos. A ferramenta foi inserida no contexto de trabalho deles para que metade dos grupos da disciplina pudesse utiliza-la para a realização do projeto final de avaliação. Parte da validação feita em sala de aula envolve a comparação dos resultados alcançados com e sem o uso da ferramenta.

A avaliação do software em relação ao usuário deve levar em conta aspectos como Eficiência, Eficácia e Satisfação (FILARDI; TRAINA, 2008). O questionário tem como propósito então avaliar se a ferramenta possibilitará aos usuários alcançar uma solução para o projeto conceitual satisfatória e de forma facilitada.

As perguntas foram direcionadas para cada um dos aspectos do software a serem testados, separadas por funcionalidade do software:

- Interface
 - Cadastro de novos casos
- Edição de casos
 - Decomposição funcional
 - Matriz Morfológica
- Importação / Exportação de casos
- Questionamentos gerais sobre a ferramenta

Ao elaborar as questões um dos cuidados foi criá-las de forma a analisar situações que não poderiam ser observadas normalmente, e dependem da percepção do usuário para serem respondidas (RUBIN, J., & CHISNELL, 2008). Houve também preocupação com a objetividade e tamanho das questões, evitando perguntas com mais de uma frase e também evitando direcionar a opinião dos usuários. O tamanho do teste em si foi levado em consideração e foi evitado um número muito grande de questões para evitar fadiga dos entrevistados mas ao mesmo tempo cobrir todos os pontos necessários, de acordo com as diretrizes de (SHARP; ROGERS; PREECE, 2007).

Foram utilizadas perguntas Escalares para julgar alguns itens mais objetivos e Abertas para colher as opiniões dos usuários quando necessário (DIX et al., 2012).

O questionário individual foi entregue a cada um dos alunos como forma de avaliar o desempenho da ferramenta no auxílio ao projeto e a sua facilidade de uso.

Idealmente, cada aluno teria que fazer o projeto conceitual utilizando somente a base com os conceitos (já cadastrados e os próprios que eles mesmos cadastrarem) e logo após repetir o processo com o auxílio da ferramenta de sugestão. Todos os casos criados deverão ser enviados para cadastro em uma base única após a finalização.

O ideal é que os alunos façam uso da ferramenta sem interferência externa, com somente uma breve explicação das funcionalidades e de seus objetivos.

O formulário de validação para os alunos encontra-se no Anexo I.

4.1.1.1 Ambiente de testes

Todos os alunos receberam uma instância do aplicativo para uso em seus projetos. Todas as instâncias tinham a mesma base de dados para avaliação da performance nos vários tipos de projetos da disciplina.

Os testes foram realizados no Laboratório de Computação da Faculdade de Tecnologia da Universidade de Brasília, aonde a ferramenta foi instalada para que os alunos pudessem se familiarizar com seu uso logo após uma breve explicação sobre o projeto, seus objetivos e a própria ferramenta.

Os questionários foram entregues após o término do uso do aplicativo (em sala de aula) e terão como objetivo avaliar a facilidade de uso, a adequação do sistema aos conceitos utilizados pelos alunos no desenvolvimento do projeto conceitual, e o impacto do uso da ferramenta na qualidade das soluções geradas.

O formulário de avaliação se encontra no Anexo II, e no Anexo V o registro fotográfico da interação com os alunos no ambiente.

4.1.2 Verificação

Dois pontos serão avaliados na verificação. Durante o desenvolvimento serão efetuados os testes descritos no

para os casos de uso levantados a partir dos requisitos do software para gerar um protótipo estável para os testes da etapa de Validação. O material gerado então pela etapa de Validação será utilizado para realizar um teste final de desempenho da funcionalidade de sugestão para verificação da correta implementação dos conceitos do RBC no sistema.

A base com todos os casos gerados pelos alunos durante o semestre será utilizada para realizar a etapa de Verificação, baseando a análise nos requisitos do software levantados no item 3.2 deste documento:

- Um projeto de referência será escolhido para cadastro antes, gerando um caso que será guardado. Neste momento o caso não será armazenado na base.
- Após o aumento do volume dos casos da base, o procedimento será repetido com as informações do mesmo caso do item acima.
- Os resultados da sugestão de funções para cada uma das etapas acima será comparado para verificar a melhoria de performance na sugestão das soluções com o aumento do volume da base.

4.1.2.1 Ambiente de testes

O ambiente de testes para a verificação do aplicativo será o mesmo de desenvolvimento.

4.2 Resultados esperados

Espera-se que

- os alunos da disciplina tenham uma relativa facilidade no uso da ferramenta, visto que a mesma é direcionada ao conteúdo lecionado neste período letivo. Uma das expectativas é a de que o processo de criação seja mais fluido devido às referências para as soluções contidas

na ferramenta, servindo para que os alunos utilizem os conceitos já cadastrados ou tenham uma base para dar início ao processo criativo. Um fator diferencial é o fato dos alunos não cursarem especificamente o curso de Engenharia Mecatrônica, mas que terão acesso às soluções já consolidadas neste domínio.

- Comparando as soluções criadas pelos alunos com a ferramenta tenha princípios de solução mais bem avaliados em relação aos que não se utilizaram do sistema. Isto será feito com conjunto com a Professora Andrea (que leciona o curso aos alunos) durante a avaliação do projeto final da disciplina.
- Volume de dados influencie positivamente na criação de novas soluções, e que o projeto cadastrado como referência tenha como sugestões de solução um conjunto mais plausível ou de melhor qualidade de acordo com o crescimento da base de referência.

A seguir veremos as análises feitas com os dados obtidos nas duas etapas.

4.3 Resultados obtidos

A apresentação dos resultados também foi dividida em duas partes, seguindo o modelo já apresentado em duas etapas para a análise da ferramenta.

4.3.1 Validação

O formulário contendo as perguntas foi enviado para todos os alunos para avaliação, e as respostas são mostradas na Tabela 4 (ilustradas na Figura 32), e a Tabela 5 mostra os projetos por eles realizados:

Tabela 4 - Resultados da avaliação em sala de aula

Item	Questão	Resultado
1	Falhas são mostradas de forma eficiente	2,9
2	Funcionalidades desejadas são encontradas com facilidade	3,2
3	Os itens de ajuda são visíveis	3,3
4	Os itens de ajuda possuem informações relevantes	3
5	A montagem dos desdobramentos das funções é intuitiva	3,6

Item	Questão	Resultado
6	Os parâmetros que descrevem as funções são pertinentes	3,4
7	A montagem da matriz morfológica é intuitiva	3,9
8	A funcionalidade de edição dos casos facilita a geração de variações para o projeto	2,7
9	Interface familiar aos conceitos de projeto de produto conceitual	3,9
10	O uso do software agilizou no desdobramento das funções	3
11	As funções foram relacionadas como semelhantes de forma correta	3
12	Os conceitos sugeridos foram úteis para o projeto	3
13	A ferramenta de sugestão foi útil ao projeto	3,1
14	O uso do software agilizou a composição da matriz	2,8
15	O cadastro de conceitos é intuitivo	3,7
16	O cadastro de conceitos é rápido	4,2
17	A interface de sugestão de conceitos é intuitiva	3,6
18	Dificuldade ao exportar os casos da base	3,8
19	Dificuldade ao importar os casos da base	3,4
20	O uso do software agilizou o projeto conceitual como um todo	3,1
21	O cadastro do caso ocorreu sem falhas	1,8
22	O usuário conseguiu cadastrar todos os casos	1,5
23	Foram necessárias mais de uma tentativa para cadastro do caso	1,3

Tabela 5 - Projetos dos grupos avaliados

1	TranCar – Sinalizador eletrônico de estado de tranca automotiva
2	Monitor de Validade de alimentos
3	Medidor de Volume para Tanque de fermentação de Cervejas
4	Marmita com função de aquecimento
5	Impressora 3D de concreto para Construção Civil

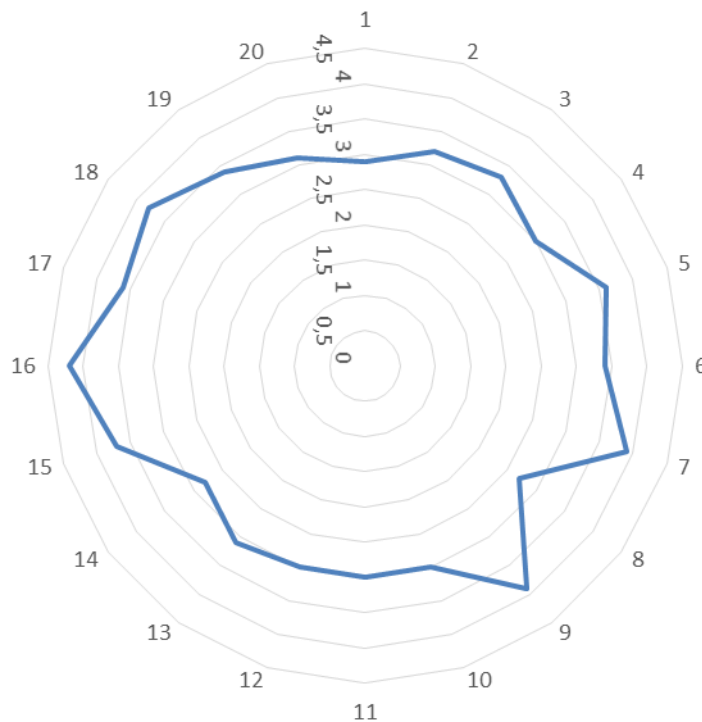


Figura 32 - Distribuição dos valores da escala por item

Para as questões de 1 a 20, a escala das respostas deve ser de 0 para pouco e 5 para muito. As questões de 21 a 23 tem com resposta 1 para sim e 2 para não. As respostas completas de cada grupo estão presentes no Anexo III.

Para este teste, era esperado que as funções já tivessem algum desempenho, e em alguns grupos as soluções sugeridas tiveram impacto na composição da matriz, mas como era esperado não existia ainda base suficientemente grande para que todas as funções pudessem ser preenchidas automaticamente.

A maior dificuldade encontrada pelos estudantes foi o intercâmbio de dados. Algumas falhas ainda presentes no software dificultaram a troca de projetos e a edição de novas variações dos casos por eles cadastrados. Todos os casos por eles cadastrados foram enviados para importação em uma única base para a realização do procedimento de validação.

Das questões objetivas, entende-se que:

- As janelas de notificação de erros foram bem sucedidas na maioria dos casos para informar falhas no correto fluxo de uso.

- A interface de cadastro teve sucesso em transparecer os conceitos necessários ao projeto conceitual.
- A reedição dos casos seria melhor aproveitada se pudesse ser feita em etapas (atualmente o caso só pode ser salvo após o cadastro completo do desdobramento funcional e da matriz morfológica)
- A funcionalidade de sugestão foi útil para alguns grupos, porém para alguns não foram encontradas relações suficientes para agilizar de forma satisfatória o projeto.
- O cadastro de conceitos foi bem utilizado e não apresentou grandes problemas nas funcionalidades presentes.
- Ainda restam algumas situações de erro que impediram o correto uso do software e impactou no desempenho do mesmo, criando dificuldades para o cadastro e edição dos casos.

Das questões discursivas, que podem ilustrar pontos mais específicos sobre a utilização do software, foram retiradas as seguintes conclusões:

- Ao selecionar a descrição das funções e a tipologia das mesmas, não houve total entendimento quanto ao objetivo da atividade. Alguns alunos entenderam que as funções selecionadas deveriam ser somente as sugeridas e outros tiveram, mesmo após a explicação, dificuldade na classificação das funções.
- Os conceitos cadastrados já deveriam estar prontos para uso, mas alguns alunos sentiram a necessidade de alterar as imagens dos conceitos para uso em seus projetos (alguns conceitos iniciais tiveram cadastrada somente a imagem padrão).
- A localização das funcionalidades teve também alguma dificuldade, porém em menor grau. As caixas de ajuda, por exemplo, que explicavam o preenchimento da caixa de funções não foi vista por alguns grupos que informaram a falta desta função.

- A interface de edição (tanto a decomposição funcional, quanto a matriz morfológica) ainda apresentou erros que não puderam ser encontrados durante os testes de validação. Algumas situações mais atípicas foram encontradas no uso de máquinas com codificação de caracteres diferenciada, que fez com que a importação e exportação dos casos fosse dificultada.

A Preocupação desta etapa era avaliar se o uso do aplicativo teria impacto no tempo levado pelos alunos para a realização do projeto conceitual e o apoio que a mesma forneceria em termos de qualidade e criatividade. Nestes termos, o impacto no projeto foi pouco porém esperado devido à quantidade de casos cadastrados ainda estar abaixo do ideal, mas mesmo com o número limitado alguns grupos puderam se beneficiar da base e do sistema de sugestão.

Dois grupos em especial tiveram bastante êxito ao utilizar a ferramenta, utilizando o sistema de sugestionamento de forma mais abrangente.

O projeto 4 – Marmita com função de aquecimento gerado pelos alunos teve três das suas funções preenchidas pelo software com sucesso, como ilustrado na Figura 33.

ligar dispositivo					
	Botão liga/desliga	Botão neon	Bater palmas	Chave liga/desliga	
desligar dispositivo					
	Botão liga/desliga	Botão neon	Bater palmas	Chave liga/desliga	
Captar energia					
	Fonte de energia externa	Bateria removível	Bateria recarregável	Energia Solar	Sem energia

Figura 33 - Conceitos sugerido para o projeto Marmita com função de aquecimento

O projeto mais beneficiado pela função de sugestão no entanto foi o projeto 3 - Medidor de Volume para Tanque de fermentação de Cervejas, aonde grande

parte das funções teve a solução gerada ou adaptada pela ferramenta, a matriz de soluções final para o projeto é mostrada na Figura 34.




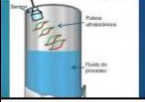




















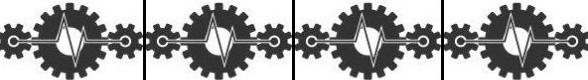












Detectar Volume							
	Sensor a Laser	balança eletrônica	Medidor Radar	Medidor de volume por ultrassom			
Detectar Temperatura							
	Termistor	Termômetro de resistência	Termômetro infravermelho	Termômetro de contato	Termômetro DIY	Termopar	Pirômetro
Detectar Pressão							
	Mostrador analógico	Componente Piezoelétrico	Transdutor de pressão	Valvula de mola			
Converter sinal analógico e...							
	Arduino	Microcontrolador	DSP	FPGA	ASIC		
processar dados							
	LabView	Microcontrolador	MatLab	Excel			
Armazenar dados e informa...							
	Hard Disk (HD)	Drive USB	Cartão SD	Armazenamento em Nuvem			
Fornecer informação							
	Antena direcional de transmissão	Cabo multicondutor	Transceptor de dados	Armazenamento em Nuvem			
Compilar informações em rel...							
	Gráfico	texto	tabelas	Relatono misto	Excel		
Prover interface do sistema							
	Display Lcd	Computador embarcado com LCD	Software proprietário				

Figura 34 - Matriz de soluções para o projeto de medição de volume de nível no taque de cerveja

Neste caso os usuários tinham pouco conhecimento técnico das soluções que poderiam ser utilizadas no projeto, e as soluções sugeridas conseguiram suprir esta deficiência.

4.3.2 Verificação

A primeira etapa da verificação foi a análise do comprimento dos requisitos funcionais realizando os testes baseados nos casos de uso documentados e percorridos com detalhes no

Anexo I : Plano de testes para o Software

. Os resultados dos testes feitos no ambiente de desenvolvimento são mostrados na Tabela 6:

Tabela 6 - Resultados dos testes com os casos de uso

ITEM	Resultado
1 - Visualização dos casos (sem busca)	ok
1.1 - Visualização dos casos (com busca / caso existente)	ok
1.2 - Visualização dos casos (com busca / caso inexistente)	ok
2 - Cadastro dos casos (Fluxo principal)	ok
2.1 - Cadastro dos casos (Sem funções elementares)	ok
2.2 - Cadastro dos casos (Erro no cadastro de função)	ok
2.3 - Cadastro dos casos (Informações de caso incompletas)	ok
2.4 - Cadastro dos casos (Sem inserção de conceitos)	ok
2.5 - Cadastro dos casos (Com cadastro de conceitos)	ok
2.6 - Cadastro dos casos (Com cadastro de conceito já existente)	ok
2.7 - Cadastro dos casos (Utilizando sugestão – funções encontradas)	ok
2.8 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)	ok
2.9 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)	ok
2.10 - Cadastro dos casos (Utilizando edição de caso já existente)	ok
3 – Importação/Exportação de informações (importação)	ok
3.1 – Importação/Exportação de informações (importação falha)	ok
3.2 – Importação/Exportação de informações (exportação)	ok

Estes testes foram bem sucedidos no ambiente de desenvolvimento, mas algumas situações foram encontradas pelos usuários e que não conseguiram ser avaliadas antes do primeiro teste, principalmente nos casos de uso relativos a importação e exportação dos casos, como já descrito na seção 4.3.1.

A última etapa dos testes do software realizada foi o teste de desempenho ao longo do tempo de vida da base de dados. Este teste foi realizado inicialmente com a base contendo os vinte e cinco casos mencionados no capítulo 3 deste documento, e o projeto analisado foi o trabalho realizado em (PÉREZ;

SANTOS, 2016), de onde foram retiradas a decomposição funcional final e a matriz morfológica geradas para a etapa de projeto conceitual do trabalho.

A análise será feita levando em conta a solução gerada pelo autor e se as soluções geradas pelo software são compatíveis com estas e também com o contexto do trabalho.

Abaixo temos na Figura 35 a decomposição funcional e na Figura 36 a matriz morfológica do projeto original para referência:

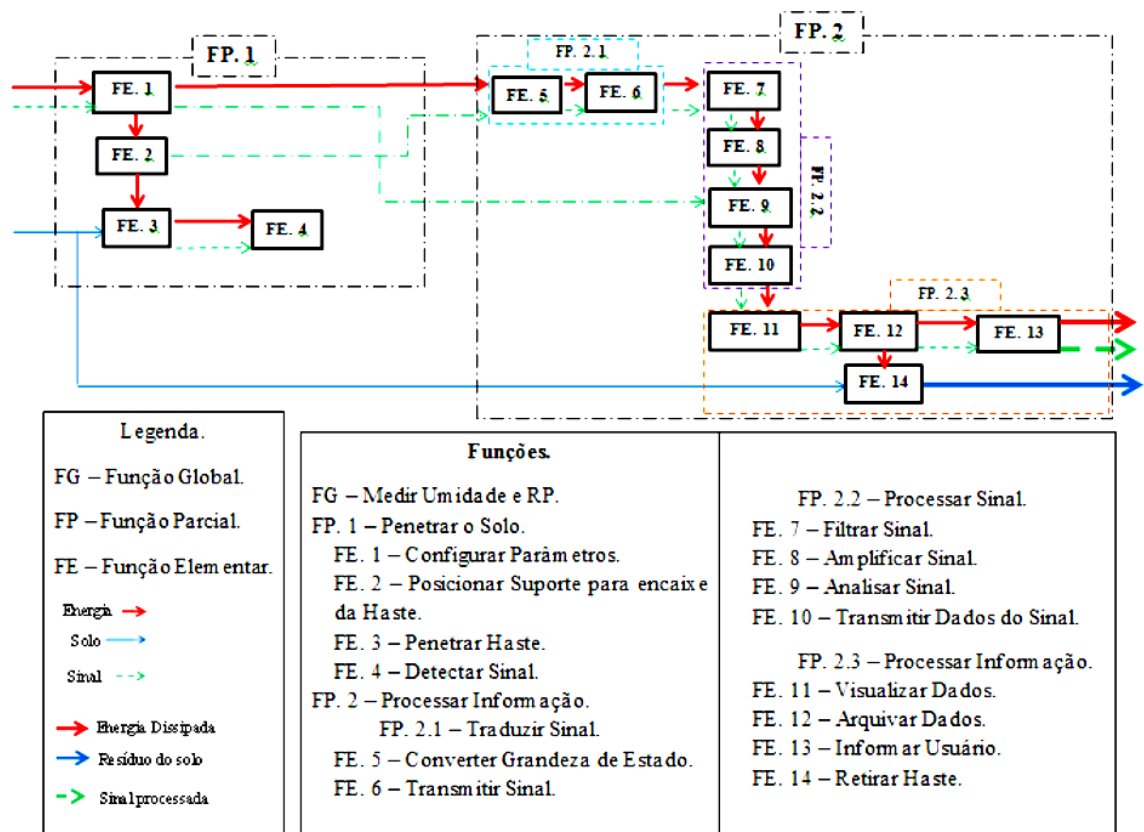








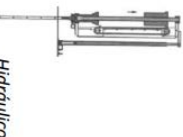

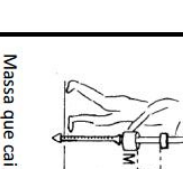











Figura 35 - Decomposição funcional (PÉREZ; SANTOS, 2016)

A geração dos conceitos para a matriz foi feita utilizando somente a funcionalidade de sugestão. Para cada uma das funções da matriz gerada na ferramenta, foi selecionada a opção de sugestão e todas as funções selecionadas foram resultado do processamento, sem o uso da opção de busca.

Para o primeiro caso, foi gerada a matriz morfológica mostrada nas Figura 37 e na Figura 38. Para o segundo caso, com a adição de 10 casos na base, temos as matrizes mostradas na Figura 39 na Figura 40.

Função	Princípios de solução			
FE. 1	 <p>Display + teclado ou botão</p>	 <p>Display 7 segmentos + teclado ou botão</p>	 <p>Interface Gráfica + Software</p>	 <p>Excel</p>
FE. 2	 <p>Manual</p>	 <p>Polias</p>	 <p>Válvula Hidráulica</p>	 <p>Motor CC</p>
FE. 3	 <p>Hidráulico</p>	 <p>Força física</p>	 <p>Massa que cai com altura constante</p>	 <p>Mecânico</p>
FE. 4	 <p>Sensores (Unidade tipo Resistivo + Impacto + ultrassônico)</p>	 <p>Sensores (Unidade tipo capacitivo + ultrassônico + força)</p>	 <p>Sensores (unidade tipo TDR+ ultrassônico) + célula de carga tipo S</p>	 <p>Sensores (unidade tipo TDR + ultrassônico + força circular tipo resistivo)</p>
FE. 5				
FE. 6	 <p>Filtro de sinal</p>	 <p>Filtro de sinal</p>	 <p>Filtro de sinal</p>	 <p>Filtro de sinal</p>
FE. 7				

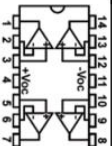
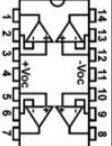
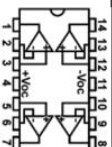
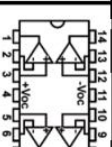

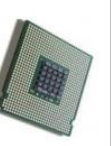

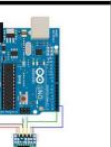







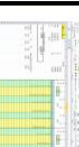




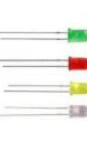



FE. 8	 <p>Amplificadores</p>	 <p>Amplificadores</p>	 <p>Amplificadores</p>	 <p>Amplificadores</p>
FE. 9	 <p>Microcontrolador</p>	 <p>Microprocessador</p>	 <p>FPGA's</p>	 <p>ARDUINO</p>
FE. 10	 <p>Conectores</p>	 <p>Conectores</p>	 <p>Conectores</p>	 <p>Conectores</p>
FE. 11	 <p>Display LCD</p>	 <p>Display 7 segmentos</p>	 <p>Interface Gráfica + Software</p>	 <p>Excel</p>
FE. 12	 <p>Pen Drive + SD card.</p>	 <p>Memoria RAM</p>	 <p>Memoria Interna</p>	 <p>Cabo USB</p>
FE. 13	 <p>Led's</p>	 <p>Buzzer</p>	 <p>Display LCD</p>	 <p>Display 7 segmentos</p>

Figura 36 - Matriz Morfológica (PÉREZ; SANTOS, 2016)









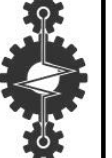







Configurar Parâmetros								
								
Posicionar suporte para enc...								
Penetrar/hasse								
Detectar Sinal				ADC	DAC	Mudança de nível	Sinal enviado por cabo	Compressão de banda
Converter grandeza de estado								
Transmitir sinal								
Filtrar sinal								

Figura 37 - Matriz gerada com a base inicial (primeira parte)







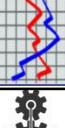
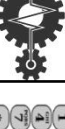


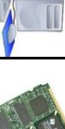
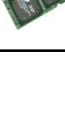












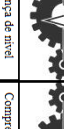











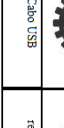
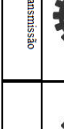

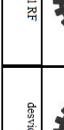



configurar parâmetros												
	OS proprietário	OS terceirizado	Página WEB	OS proprietário com Base de dados	OS terceirizado com Base de dados	Encoder rotativo	Ajuste por gráfico	Inteligência artificial	Teclado numérico	nível de base anteriormente avaliado	Entrada de dados via rede	DSP
Pesquisar suporte para enc...												
	Cinto	Rede	Banco	Suporte Vertical								
Repetir haste												
	nada											
Detectar sinal												
	Sinal enviado por cabo	Sinal enviado por RF	Software proprietário	Hardware	NFC	Hierdo Software/hardware	DAC	Mudança de nível	Compressão de banda	áudio	WiFi	ADC
Converter geradora de estado												
	nada											
Transmitir sinal												
	Sinal enviado por RF	Software proprietário	Hardware	I2C	Hierdo Software/hardware	SPI	Cabo USB	retransmissão	CAN	Sinal RF	desvio ativo	sem comunicação
Filtrar sinal												
	nada											

Figura 39 - Matriz gerada com a base final (parte 1)






























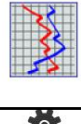




































Amplificador serial															
	tada														
analisar sinal			Computador de placa única				PLC		SPI		CAN				
Transmitir dados do sinal			Software proprietário				Híbrido Software/hardware								
Visualizar dados			audio				Gráficos								
Arquivar dados			Drive USB				Arquivo xml								
Informe usuário			audio				WiFi								
Relevar hardware			tada												

Figura 40 - Matriz gerada com a base final (parte 2)

Pode-se observar que a lógica esperada do sistema foi a apresentada nas matrizes geradas. Na análise quantitativa, uma maior quantidade de funções conseguiu ser preenchida através do uso da funcionalidade de sugestão, ao mesmo tempo em que as que já existiam tiveram mais conceitos associados como solução.

Algumas funções também tiveram soluções compatíveis com as geradas no trabalho:

- configurar parâmetros
- analisar sinal
- arquivar dados
- visualizar dados
- informar usuário

Todas estas tiveram conceitos sobrepostos, que foram selecionados no trabalho e ao mesmo tempo sugeridos durante o uso da ferramenta.

4.4 Considerações finais

A etapa de testes foi executada com êxito e o seu correto planejamento mostrou-se uma ferramenta importante de apoio ao ensino de projeto de produtos.

As avaliações realizadas mostraram vários dados importantes relacionados tanto à robustez do software quanto a sua utilidade como ferramenta de auxílio ao projeto. Estes dados servirão como ponto de apoio para futuros desenvolvimentos, tanto em novas funcionalidades quanto na melhoria da ferramenta.

Ao final desta avaliação, a base de dados de projeto conta com 36 projetos cadastrados, uma versão alfa da ferramenta disponível para uso acadêmico.

5 Conclusão

Neste capítulo serão discutidas as considerações finais do projeto e avaliado o cumprimento dos objetivos iniciais da pesquisa.

5.1 Análise de objetivos e resultados

A avaliação final mostrou resultados promissores em relação ao seu uso por parte dos alunos, mantendo aberta a possibilidade de transformar esta ferramenta em um projeto complexo envolvendo outras atividades da fase de projeto de produto conceitual.

O comportamento do sistema em relação ao crescimento da base de dados correspondeu ao esperado, com soluções mais completas surgindo com o aumento do número de casos. A possibilidade de alterar os algoritmos e implementar modelos com melhor funcionalidade ao projeto está presente na arquitetura do projeto, deixando o espaço aberto para novas implementações. Trabalhos como (AAMODT; PLAZA, 1994), (EL-SAPPAGH; ELMOGY, 2015) e (RICHTER; AAMODT, 2005) mostram que RBC é um conceito que ainda mostra potencial.

Constatou-se que o reuso das soluções necessita de maior refinamento, e que técnicas mais avançadas poderiam ser utilizadas para classificar melhor os itens sugeridos. O foco do desenvolvimento foi voltado em grande parte à estruturação e análise de similaridade dos casos, o que afetou a busca por melhores soluções para o reuso de conceitos.

A etapa de testes realizada, levando em contas os conceitos propostos por (SOMMERVILLE, 2010) e (FDA, 2002), mostra que os testes de sistemas devem ser planejados de forma clara e organizada para que se tenha o devido retorno dos usuários e para que seja útil ao desenvolvimento.

As avaliações e as experiências dos alunos ao fim dos testes foram bastante positivas, recebendo um retorno amigável quanto ao seu uso e ao desempenho das funcionalidades disponíveis e também das falhas e melhorias que somente o retorno do usuário pode prover.

Cumpridos os objetivos iniciais da pesquisa, todos os nossos esforços conseguiram se materializar em uma ferramenta utilizável, porém, observa-se também que há bastante espaço para melhora e desenvolvimento de novas funcionalidades e que novos estudos ainda podem ser feitos para levar adiante a ferramenta e os conceitos nela aplicados.

5.2 Limitações da pesquisa

O maior desafio enfrentado nesta pesquisa foi a busca por informações de projeto que pudessem servir para popular a base de casos. A quase totalidade dos projetos cadastrados na base atual é de fonte acadêmica, sendo assim, serve como prova de conceito.

Os testes realizados com os alunos conseguiram esclarecer muitos dos pontos relacionados ao modo como os conceitos empregados na ferramenta funcionariam em um ambiente de projeto, porém se faz necessário um teste com projetistas mais experientes e com uma versão mais avançada da aplicação.

O tempo disponível para a etapa de desenvolvimento e validação do projeto foi limitado, idealmente teríamos mais um ciclo de correções e ajustes nas funcionalidades para proposição de uma ferramenta mais robusta.

5.3 Recomendações para futuros trabalhos

O projeto teve algumas concepções que foram anotadas para serem tomadas como referência para novas pesquisas:

- Elementos colaborativos - Este é um ponto que foi considerado no começo do projeto, mas que foi deixado como opcional. A ferramenta se beneficiaria enormemente de um servidor central que conseguisse

concentrar as informações dos projeto, tornando mais fácil e dinâmico o intercâmbio de informações entre os projetistas.

- Classificação das soluções para melhor avaliação de reuso - As soluções podem ter a possibilidade de ser classificadas após a fase do projeto conceitual ou mesmo integrando ao projeto uma etapa a mais para a seleção da solução final, gerando uma pontuação para cada solução baseada no seu contexto de uso, melhorando a qualidade das soluções apresentadas.
- Sugestões de desdobramentos de funções - As funções apresentadas possuem uma forma de classificação e meios de avaliar as semelhanças entre elas. Outra alternativa seria avaliar se o sistema da forma como se encontra pode ser adaptado para, não só fornecer indicações de solução para cada função, incluir também avaliação dos modos de desdobramento de uma função baseados na sua classificação.
- Diferentes modelos de classificação e armazenamento das funções - O modelo de classificação das funções aqui apresentado serviu aos propósitos da ferramenta. Observa-se que devida a sua complexidade e quantidade de parâmetros necessários para a sua classificação sugere-se a proposição de um modelo que diminua o tempo de classificação das funções.

REFERÊNCIAS BIBLIOGRÁFICAS

AAMODT, A.; PLAZA, E. Case-based reasoning: Foundational issues, methodological variations, and system approaches. **AI communications**, v. 7, n. 1, p. 39–59, 1994.

ADAMSSON, N. Mechatronics engineering New requirements on cross-functional integration. p. 60, 2005.

AGRES, K.; WIGGINS, G. A. Schematic processing as a framework for learning and creativity in CBR and CC. **CEUR Workshop Proceedings**, v. 1520, p. 151–155, 2015.

APACHE FOUNDATION. **Apache Maven**. Disponível em: <<https://maven.apache.org/>>. Acesso em: 10 out. 2016.

BACK, N.; OGLIARI, A.; DIAS, A.; SILVA, J. C. **Projeto integrado de produtos: planejamento, concepção e modelagem**. Barueri, SP: Manole, 2008.

BAQAPURI, A. I. Deep Learning Applied to Image and Text Matching. 2015.

BAXTER, M. Projeto de Produto: guia prático para o design de novos produtos. **Editores Blucher**, 2000.

BROWNLEE, J. **What is Machine Learning: A Tour of Authoritative Definitions and a Handy One-Liner You Can Use**. Disponível em: <<http://machinelearningmastery.com/what-is-machine-learning/>>.

BROWNLEE, J. **Process for working through Machine Learning Problems**. Disponível em: <<http://machinelearningmastery.com/process-for-working-through-machine-learning-problems/>>. Acesso em: 29 abr. 2016.

CATECATI, T. et al. Métodos Para a Avaliação da Usabilidade no Design de Produtos. **DAPesquisa**, v. 4, n. 8, p. 564–581, 2011.

CHANDRASEGARAN, S. K. et al. The evolution, challenges, and future of knowledge representation in product design systems. **Computer-Aided**

Design, v. 45, n. 2, p. 204–228, 2013.

CHEN, Y.; LIU, Z. L.; XIE, Y. B. A knowledge-based framework for creative conceptual design of multi-disciplinary systems. **CAD Computer Aided Design**, v. 44, n. 2, p. 146–153, 2012.

DEEPLARNING4J. **Word2Vec**. Disponível em: <<https://deeplearning4j.org/word2vec>>. Acesso em: 25 nov. 2016.

DENG, L. Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. **Research.Microsoft.Com**, 2013.

DIX, A. et al. **HUMAN-COMPUTER INTERACTION**. 3. ed. [s.l.] Prentice - Hall, 2012. v. 1

DOMINGOS, P. A few useful things to know about machine learning. **Communications of the ACM**, v. 55, n. 10, p. 78, 2012.

ECLIPSE FOUNDATION. **Eclipse Luna**. Disponível em: <<https://eclipse.org/luna/>>. Acesso em: 10 out. 2016.

EL-SAPPAGH, S. H.; ELMOGY, M. Case Based Reasoning: Case Representation Methodologies. **International Journal of Advanced Computer Science and Applications,(IJACSA)**, v. 6, n. 11, p. 192–208, 2015.

FARGNOLI, M.; ROVIDA, E.; TROISI, R. The morphological matrix: Tool for the development of innovative design solutions. **Proc. ICAD**, p. 1–6, 2006.

FDA. General Principles of Software Validation ; Final Guidance for Industry and FDA Staff. **Health San Francisco**, p. 47, 2002.

FILARDI, A. L.; TRAINA, A. J. M. Montando questionários para medir a satisfação do usuário: Avaliação de interface de um sistema que utiliza técnicas de recuperação de imagens por conteúdo. **Ihc 2008**, p. 176–185, 2008.

FINNIE, G. Similarity and metrics in case-based reasoning. n. August 2016, 2002.

GITHUB INC. **GitHub**. Disponível em: <<https://github.com/>>. Acesso em: 10 out. 2016.

HEHENBERGER, P.; ZEMAN, K. Design activities in the development process of mechatronic systems. **2007 IEEE/ASME international conference on advanced intelligent mechatronics**, p. 1–6, 2007.

HIPP, R.; DAN KENNEDY; MISTACHKIN, J. SQLITE. 2016.

LEON, N. The future of computer-aided innovation. **Computers in Industry**, v. 60, n. 8, p. 539–550, 2009.

LOPEZ DE MANTARAS, R. et al. Retrieval, reuse, revision and retention in case-based reasoning. **The Knowledge Engineering Review**, v. 20, n. 3, p. 215, 2006.

LOPEZ FLORES, R. et al. Collective intelligence to solve creative problems in conceptual design phase. **Procedia Engineering**, v. 131, p. 850–860, 2015.

MAIN, J.; DILLON, T.; SHIU, S. A Tutorial on Case-Based Reasoning. p. 1–28, 2001.

MITCHELL, T. M. The Discipline of Machine Learning. **Machine Learning**, v. 17, n. July, p. 1–7, 2006.

ORACLE. **Oracle Java**. Disponível em: <<https://www.oracle.com/br/java>>. Acesso em: 10 out. 2016.

PÉREZ, M. M.; SANTOS, A. C. DOS. **PROJETO CONCEITUAL DE UM INSTRUMENTO PARA AVALIAR O ESTADO DE COMPACTAÇÃO DO SOLO**. [s.l.] Universidade de Brasília, 2016.

RED HAT; JBOSS DEVELOPER. **Hibernate**. Disponível em: <<http://hibernate.org/>>.

RICHTER, M. M.; AAMODT, A. Case-based reasoning foundations. **The Knowledge Engineering Review**, v. 20, n. 1983, p. 203, 2005.

RIT. **EDGE**. Disponível em: <<http://edge.rit.edu/edge/Welcome/public/Home>>.

ROBERTSON, B. F.; RADCLIFFE, D. F. Impact of CAD tools on creative problem solving in engineering design. **CAD Computer Aided Design**, v. 41, n. 3, p. 136–146, 2009.

ROBLES, G. C. et al. Case-Based Design and TRIZ: a knowledge based

design process. **Design**, p. 1–7, 2010.

ROZENFELD, H. et al. **Gestão do Processo de Desenvolvimento de Produtos: uma referência para a melhoria do processo**. 1. ed. São Paulo, SP: Saraiva, 2006.

RUBIN, J., & CHISNELL, D. **Handbook of usability testing [electronic resource]: How to plan, design, and conduct effective tests (2nd ed.)**. [s.l: s.n.].

RZEVSKI, G. On conceptual design of intelligent mechatronic systems. **Mechatronics**, v. 13, n. 10, p. 1029–1044, 2003.

SELL, R.; TAMRE, M. Mechatronics System Design Process and Methodologies. n. August, 2016.

SHARP, H.; ROGERS, Y.; PREECE, J. Interaction design: beyond human-computer interaction. **Book**, v. 11, p. 773, 2007.

SOMMERVILLE, I. **Software Engineering**. [s.l: s.n.].

TEIXEIRA, F. G.; RÉGIO PIERRE DA SILVA; BRUNO, F. B. Um Sistema WEB para o Projeto Conceitual de Produtos. n. April 2016, p. 1222–1227, 2008.

VEGTE, W. F. VAN DER; BREEMEN, E. J. J. Flowchart-Assisted Function Analysis of Products to Support Teaching of The Exact Sciences. **International Conference on Engineering Design, ICED**, p. 101–112, 2009.

WANG, L. et al. Collaborative conceptual design - State of the art and future trends. **CAD Computer Aided Design**, v. 34, n. 13, p. 981–996, 2002.

WIKIMEDIA. **Wikidata:Database download/pt-br**. Disponível em: <https://www.wikidata.org/wiki/Wikidata:Database_download/pt-br>. Acesso em: 25 ago. 2016.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining: Practical Machine Learning Tools and Techniques, Third Edition**. [s.l: s.n.], v. 54

YAN, Y.; CHILDS, P. R. N. Creativity tool selection for design engineers in idea generation. **International Conference on Engineering Design**, n. July, p. 1–12, 2015.

ZANNI-MERK, C.; CAVALLUCCI, D.; ROUSSELOT, F. An ontological basis for computer aided innovation. **Computers in Industry**, v. 60, n. 8, p. 563–574, 2009.

ZHENG, C. et al. Mechatronic Design Process: A Survey of Product Data Model. **Procedia 24th CIRP Design Conference**, v. 21, n. 2, p. 282–287, 2014.

ZHU, Z. et al. Literature review on the creativity of CBR applications. **Artificial Intelligence Review**, v. 40, n. 4, p. 379–390, 2013.

ANEXOS

		Pág.
Anexo I	Plano de testes para o Software	107
Anexo II	Formulário de avaliação de uso	107
Anexo III	Respostas dos grupos ao teste de uso	107
Anexo IV	Respostas das questões discursivas	109
Anexo V	Testes com os grupos selecionados	112
Anexo VI	Endereço do repositório GitHub	113

Anexo I : Plano de testes para o Software

Ver documento a parte

Anexo II : Formulário de avaliação de uso

Ver documento a parte

Anexo III : Respostas dos grupos ao teste de uso (Objetivas)

1	TranCar – Sinalizador eletrônico de estado de tranca automotiva
2	Monitor de Validade de alimentos
3	Medidor de Volume para Tanque de fermentação de Cervejas
4	Marmita com função de aquecimento
5	Impressora 3D de concreto para Construção Civil

Questão	1	1	1	2	2	2	2	2	2	3	3	4	4	4	5	Resultado
Falhas são mostradas de forma eficiente	3	1	3	5	3	2	3	3	3	2	3	3	2	2	5	2,9
Funcionalidades desejadas são encontradas com facilidade	4	3	3	4	2	3	2	4	3	3	4	2	2	4	4	3,2
Os itens de ajuda são visíveis	2	2	3	4	5	3	4	4	2	3	4	2	2	4	5	3,3
Os itens de ajuda possuem informações relevantes	2	3	3	4	3	2	3	3	3	4	3	2	2	4	4	3
A montagem dos desdobramentos das funções é intuitiva	4	3	1	3	2	4	3	3	3	5	5	5	4	4	5	3,6
Os parâmetros que descrevem as funções são pertinentes	3	4	2	3	4	2	2	3	4	4	4	4	4	4	3	3,4
A montagem da matriz morfológica é intuitiva	3	4	4	4	2	4	4	4	4	4	4	4	3	5	5	3,9
A funcionalidade de edição dos casos facilita a geração de variações para o projeto	3	2	5	5	1	1	3	4	3	2	3	1	4	3	0	2,7
Interface familiar aos conceitos de projeto de produto conceitual	3	4	5	4	3	4	4	3	3	4	5	5	4	3	4	3,9

O uso do software agilizou no desdobramento das funções	3	1	3	3	1	2	2	4	1	4	5	3	4	5	3	3
As funções foram relacionadas como semelhantes de forma correta	1	4	3	2	3	4	2	3	2	4	4	3	2	4	3	3
Os conceitos sugeridos foram úteis para o projeto	2	3	3	3	2	1	2	4	1	5	5	2	5	4	2	3
A ferramenta de sugestão foi útil ao projeto	2	3	3	3	1	2	3	3	1	5	5	3	3	5	4	3,1
O uso do software agilizou a composição da matriz	1	2	3	4	1	1	2	4	2	4	5	3	4	5	0	2,8
O cadastro de conceitos é intuitivo	3	3	4	5	4	3	4	3	4	3	3	4	3	4	5	3,7
O cadastro de conceitos é rápido	4	4	4	5	5	3	4	4	4	5	3	4	3	5	5	4,2
A interface de sugestão de conceitos é intuitiva	3	3	3	4	3	4	3	4	3	4	4	4	3	4	5	3,6
Dificuldade ao exportar os casos da base	5	5	5	3	3	3	3	2	3	4	4	5	5	2	5	3,8
Dificuldade ao importar os casos da base	3	5	3	3	5	3	3	2	3	3	2	5	4	2	5	3,4
O uso do software agilizou o projeto conceitual como um todo	1	2	3	4	1	2	2	4	3	4	4	3	3	5	5	3,1
O cadastro do caso ocorreu sem falhas	2	2	2	2	2	1	2	1	1	2	2	2	2	1	2	1,8
O usuário conseguiu cadastrar todos os casos	2	2	2	2	2	1	2	1	1	1	2	1	1	1	1	1,5
Foram necessárias mais de uma tentativa para cadastro do caso	1	1	1	1	1	2	1	2	1	1	1	2	1	1	2	1,3

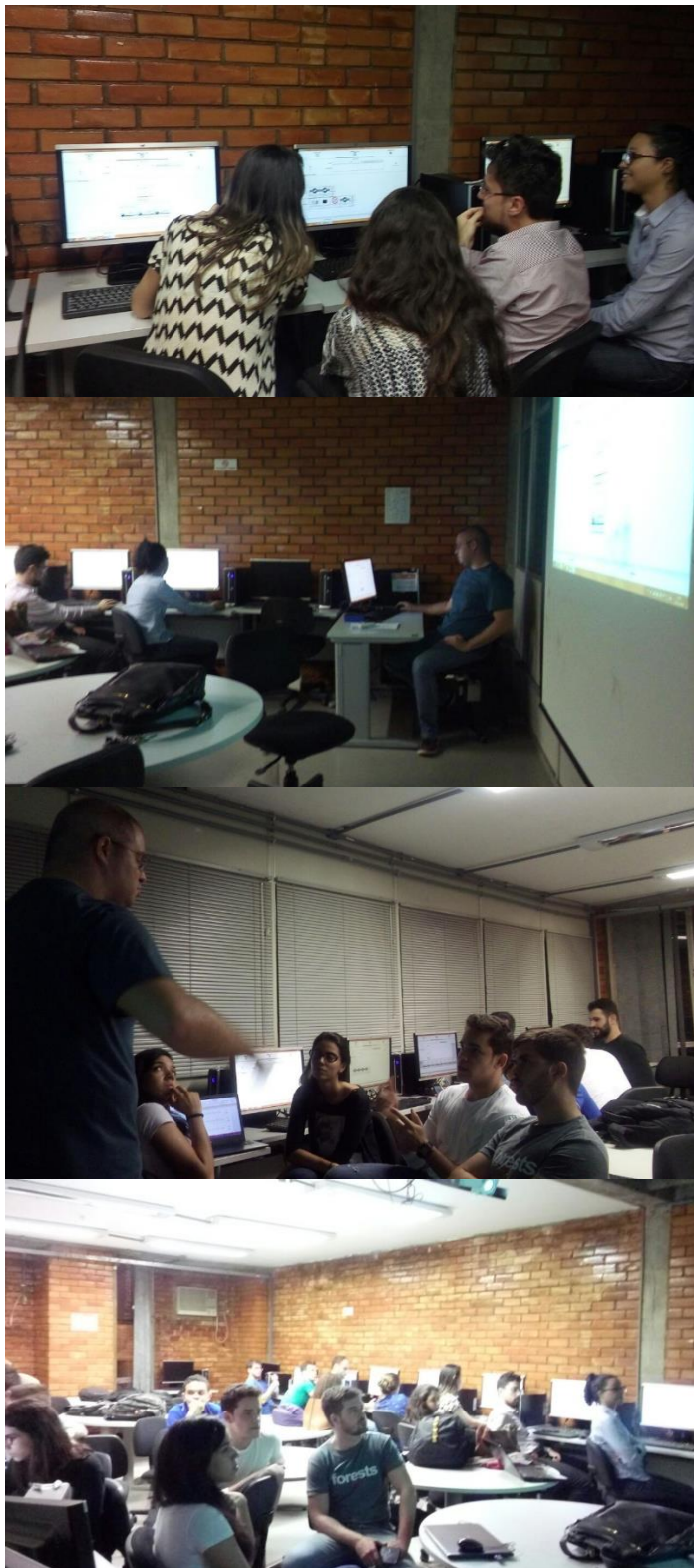
Anexo IV : Respostas dos grupos ao teste de uso (Objetivas)

Proj	Qual sua maior dificuldade no uso do software	Indique um (ou mais) ponto(s) de melhora da ferramenta avaliados a partir do seu uso	Caso tenha encontrado uma situação de erro que impediu o seu uso, descreva o ocorrido
1	<p>Salvar os casos.</p> <p>Tivemos dificuldade em salvar o trabalho realizado (modelagem funcional e matriz morfológica) e não conseguimos identificar o problema.</p> <p>Tivemos certa dificuldade também em escolher o verbo correto para algumas funções da modelagem. Por ex: algumas opções aparecem "duplicadas" na lista de opções a serem inseridas. Na verdade o que as difere, apesar de estarem escritas exatamente da mesma forma, são os parâmetros, mas isto não é claro na hora. Outras vezes ficamos em dúvida quanto à diferença de um verbo para ou o outro e, conseqüentemente, qual deveríamos utilizar.</p> <p>Após já ter criado uma matriz morfológica, uma alteração de função na modelagem funcional fazia com que a antiga funcional sumisse da matriz morfológica mas não fosse substituída pela nova.</p>	<p>Melhor a interface.</p> <p>Algumas funções já sugeridas podiam ser inseridas como compilar ou consolidar, enviar ou exportar, analisar e avaliar, entre outros.</p>	<p>Não conseguimos salvar o caso.</p> <p>Tivemos problemas para salvar o nosso projeto</p> <p>Apesar dos problemas enfrentados e registrados aqui no questionário como oportunidade de melhoria, gostaríamos de parabenizar o João pelo trabalho, pois nos ajudou MUITO na parte referente às funções que envolviam dados/informações e temos convicção de que também foi uma importantíssima ferramenta para os demais grupos devido ao número de ideias e sugestões que o programa gera, e a tendência é que isto aumente.</p> <p>Acreditamos que o uso da ferramenta será benéfico para os próximos semestres.</p>
4	<p>Acho que a maior dificuldade foi interface, que não é muito amigável. Muitas vezes fazíamos as coisas por tentativa e erro.</p>	<p>Quando passar o mouse em cima alguma configuração, aparecer uma informação rápida sobre ela.</p>	<p>Quando adicionávamos uma função na árvore, ela não aparecia na matriz morfológica, apesar de outras estarem lá. Tivemos que adicioná-la duas vezes para que uma aparecesse.</p>
2	<p>preencher os quadros na matriz morfológica.</p>	<p>cada opção pode ser melhor descrita, alguns símbolos podem ter os seus significados explicados.</p>	-
3	<p>Salvar e exportar a matriz final.</p>	<p>Edição de conceitos já existentes e melhora na hora de salvar.</p>	<p>Não foi possível salvar, sempre apontava erro.</p>

4	<p>No projeto Marmita tivemos que cadastrar vários conceitos, e primeiro fizemos o cadastro do nome, achando que a imagem poderia ser adicionada posteriormente, o que não foi possível. Assim o grupo teve que recadastrar os mesmos conceitos, para poder adicionar a imagem. Outra dificuldade enfrentada foi que o programa, após salvar o caso, excluiu alguns conceitos definidos para as funções na matriz morfológica, sendo que esses ficaram vazios, e o caso tinha que ser editado novamente, o que gerou outra dificuldade: na árvore de funções para editar o caso, ele exclui todas as classificações que foram realizadas, e é necessário fazer todas, em todas as funções, novamente.</p>	Melhorar os itens apontados na dificuldade de uso do sistema.	ND.
2	<p>Tive algumas dificuldades para entender inicialmente o que significava cada símbolo dentro da caixa de funções, e para que eles servem.</p>	<p>Seria interessante ter uma caixa texto explicando como usar o programa ou que significa cada símbolo, até mesmo um link explicando os passos para usar o programa como uma apostila de usuário.</p>	<p>Não encontrei nenhuma situação de erro</p>
2	<p>O software sugeriu poucas sugestões de funções que poderiam ser aplicadas ao nosso produto.</p>	<p>O sistema deve poder permitir que o usuário adicione as imagens das funções criadas depois delas serem adicionadas, e não, ao mesmo tempo.</p>	<p>Tentamos adicionar as imagens alguns dias depois de termos adicionado as funções. Não foi possível e por isso tivemos que refazer o que já tínhamos feito, instalando e reinstalando o software várias vezes. Muitas vezes não conseguimos abrir o software</p>
2	<p>Entender os tipos de funções e por ser um projeto específico tivemos que incluir todas nossas opções no programa.</p>	<p>Ser mais intuitiva, ter opção de editar os conceitos</p>	<p>Não houve</p>
4	<p>Ele é diferente de todos os softwares que já utilizei. Demorei um pouquinho no começo, mas depois da explicação consegui utiliza-lo bem.</p>	<p>Aumentar o banco de dados de funcionalidades. Fora isso, achei o software muito bom.</p>	<p>Não se aplica.</p>

2	Edição para adicionar imagens para os princípios de solução já existentes, as vezes não consegui abrir o software, não achei intuitiva a maneira de encontrar matrizes (projetos) já cadastrados, talvez tenha sido pelo programa ter dado erro.	Torná-lo mais intuitivo e mais fácil de abri-lo e configurá-lo.	Erro na hora de reabrir o projeto, parou de funcionar e não permitia a alteração do projeto.
2	Base de dados ainda está um pouco incompleta.	Edição de princípios de solução para inserir imagem sem ter que criar novamente.	-
1	Salvamento/Exportação	Salvamento/Exportação	Salvamento do caso apenas depois da conclusão do mesmo, o que possibilita perda de todos os dados já adicionados anteriormente
1	Exportar o documento, Salvar o documento, Editar Imagens dos Conceitos já existentes, Salvar caso na Base	Salvamento dos casos - muita demora. Sugestão de conceitos - sugeridos conceitos que não estavam relacionados.	Tentou-se salvar o arquivo, porém, o programa ficou por 40min rodando as engrenagens, bloqueado. O programa foi fechado, 3 o caso já estava salvo na base. Logo, houve problema no retorno do salvamento ao programa, pois o programa salvou e ficou travado.

Anexo V : Testes com os grupos selecionados



Anexo VI : Endereço do repositório GitHub

URL: <https://github.com/jmtstorres/CBRMeca>

Usuário: jmtstorres

UNIVERSIDADE DE BRASÍLIA

Plano de testes

Proposta do emprego de aprendizado de máquinas como ferramenta de apoio a fase de projeto conceitual

João Marcelo Torres

13/11/2016

Descrição do planejamento e execução dos testes a serem efetuado para avaliação da ferramenta criada para o mestrado do Aluno João Marcelo Torres

Sumário

Introdução.....	3
Objetivos.....	3
Atores envolvidos	3
Requisitos da ferramenta	4
Plano de testes.....	5
Casos de uso e testes de erro	5
1 - Visualização dos casos (sem busca).....	5
1.1 - Visualização dos casos (com busca / caso existente).....	5
1.2 - Visualização dos casos (com busca / caso inexistente)	6
2 - Cadastro dos casos (Fluxo principal).....	6
2.1 - Cadastro dos casos (Sem funções elementares).....	7
2.2 - Cadastro dos casos (Erro no cadastro de função)	7
2.3 - Cadastro dos casos (Informações de caso incompletas)	8
2.4 - Cadastro dos casos (Sem inserção de conceitos).....	8
2.5 - Cadastro dos casos (Com cadastro de conceitos)	9
2.6 - Cadastro dos casos (Com cadastro de conceito já existente).....	10
2.7 - Cadastro dos casos (Utilizando sugestão – funções encontradas).....	10
2.8 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)	11
2.9 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)	12
2.10 - Cadastro dos casos (Utilizando edição de caso já existente)	12
3 – Importação/Exportação de informações (importação)	13
3.1 – Importação/Exportação de informações (importação falha)	13
3.2 – Importação/Exportação de informações (exportação)	13
Teste de Desempenho	14
Análise do uso (teste de aceitação)	15
Considerações finais	17

Introdução

A ferramenta objeto para os testes contidos neste planejamento foi elaborada para uso na dissertação de mestrado do aluno João Marcelo Torres. O projeto consiste de uma ferramenta capaz de sugerir soluções na forma de conceitos atribuídos a cada uma das funções elementares de um desdobramento funcional para a etapa conceitual do projeto de um produto mecatrônico.

Para alcançar este objetivo, foram estudados conceitos de aprendizado de máquina para poder gerar um conjunto efetivo de sugestões. Nos estudos realizados (explicados com mais clareza no documento da dissertação), a solução escolhida foi a utilização dos conceitos do Raciocínio Baseado em Casos (RBC), que compreende uma forma de utilizar projetos já realizados e documentados para a elaboração de novas soluções para o problema em questão.

Objetivos

Tratando-se o RBC de uma forma de aprendizado relacionada a experiências anteriores de projeto e tratando-se de uma ferramenta para uso em ambiente acadêmico, espera-se avaliar o software de três formas:

- Quanto a consistência do desenvolvimento, avaliando os casos de uso mais comuns
- Quanto à capacidade da ferramenta de facilitar a etapa de projeto conceitual no ambiente acadêmico
- Quanto a desempenho da ferramenta ao longo do tempo de acordo com o crescimento e revisão da base de dados.

Cada uma destas formas terá sua etapa própria de avaliação, sendo a segunda realizada com os Alunos da Professora Andrea Santos em laboratório e as outras em ambiente de desenvolvimento avaliando um caso cadastrado em dois momentos diferentes do ciclo de vida da base de dados.

Atores envolvidos

Os testes de erro e desempenho serão realizados pelo próprio desenvolvedor (na falta de recursos próprios para testes, que seria a situação ideal), e pelos alunos da turma de projeto

de produto da professora Andrea Santos, que contribuirão utilizando a ferramenta na etapa conceitual do seu projeto de conclusão de disciplina.

Requisitos da ferramenta

Nas etapas iniciais do projeto, foi elaborada uma lista de requisitos funcionais para descrever o comportamento desejado da ferramenta:

- Cadastro dos casos
 - Cadastrar novos casos
 - Editar casos já inseridos
- Interface familiar aos utilizadores da área da Engenharia de Produção
 - Decomposição funcional que mostre de forma sucinta os desdobramentos de funções e permita rápida edição dos parâmetros das funções e seus desdobramentos
 - Matriz morfológica que permita a inserção dos conceitos:
 - Automaticamente, com a funcionalidade de sugestão
 - Manualmente, a partir de:
 - Conceitos previamente armazenados
 - Novos conceitos cadastrados imediatamente
- Prover informações sobre os casos cadastrados de forma rápida e simples
- Fornecer a solução de modo legível e permitir alterações prévias ao armazenamento de um novo caso
- Permitir a exportação da base para outros sistemas e importação de casos de outras bases (desejável)
- Arquitetura modular para permitir o uso das funções do motor em sistemas externos

Os requisitos têm como foco principal poder abordar a teoria (RBC) e prover funcionalidade e facilidade de uso ao público-alvo (alunos). Alguns dos requisitos foram pensados para deixar alguma margem para melhorias, como a generalização do motor de sugestão,

permitindo que na medida em que novos algoritmos melhores sejam implementados, possam ser usados pelo aplicativo alterando somente a biblioteca do motor.

Plano de testes

O plano de testes aqui descrito terá três etapas:

- Casos de uso e testes de erro
- Testes de desempenho
- Análise de uso (testes de aceitação)

Cada uma delas é detalhada logo abaixo.

Casos de uso e testes de erro

As funcionalidades básicas do aplicativo podem ser descritas como:

- Visualização dos casos
- Cadastro dos casos
- Importação/exportação de informações

Para cada uma delas será gerado um fluxo principal e os fluxos alternativos cabíveis para o teste de erros em ambiente de desenvolvimento. Os casos gerados são listados na seção a seguir.

1 - Visualização dos casos (sem busca)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos
- **Descrição da funcionalidade** – Mostrar uma lista com todos os casos cadastrados / prover funcionalidade de busca
- **Condições Prévias** – existir casos cadastrados
- **Fluxo de ações esperado**
 - Selecionar item da lista
- **Resultado esperado** - Mostrar dados nas abas de decomposição funcional e matriz morfológica

1.1 - Visualização dos casos (com busca / caso existente)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos
- **Descrição da funcionalidade** – Mostrar uma lista com todos os casos cadastrados / prover funcionalidade de busca
- **Condições Prévias** – existir casos cadastrados com o termo de busca

- **Fluxo de ações esperado**
 - Digitar termos de busca
 - Selecionar item da lista
- **Resultado esperado** - Mostrar dados nas abas de decomposição funcional e matriz morfológica

1.2 - Visualização dos casos (com busca / caso inexistente)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos
- **Descrição da funcionalidade** – Mostrar uma lista com todos os casos cadastrados / prover funcionalidade de busca
- **Condições Prévias** – não existir casos cadastrados com o termo de busca
- **Fluxo de ações esperado**
 - Digitar termos de busca
- **Resultado esperado** - Mostrar lista vazia

2 - Cadastro dos casos (Fluxo principal)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções (independe da maneira).
 - Salvar o caso, escolhendo um local para gravar o arquivo descritivo do caso.
- **Resultado esperado** – Caso cadastrado com sucesso

2.1 - Cadastro dos casos (Sem funções elementares)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Não marcar a caixa de seleção
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
- **Resultado esperado** - Mostrar mensagem de erro indicando que não existem funções elementares selecionadas e continuar na aba de funções

2.2 - Cadastro dos casos (Erro no cadastro de função)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Não preencher os dados de classificação, o ícone no topo deve se manter com o alerta, indicando que a função está incompleta.
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica

- **Resultado esperado** - Mostrar mensagem de erro indicando que alguma das funções não está completamente preenchida e continuar na aba de funções

2.3 - Cadastro dos casos (Informações de caso incompletas)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Não inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções (independe da maneira).
 - Salvar o caso, escolhendo um local para gravar o arquivo descritivo do caso.
- **Resultado esperado** - Mostrar mensagem ao usuário informando a falta de informações do caso

2.4 - Cadastro dos casos (Sem inserção de conceitos)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa

- Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
- Alterar para a aba da matriz morfológica
- Não preencher a fila de conceitos para cada uma das funções.
- Salvar o caso, escolhendo um local para gravar o arquivo descritivo do caso.
- **Resultado esperado** - Mostrar mensagem informando que devem ser inseridos conceitos para as todas as funções presentes na matriz

2.5 - Cadastro dos casos (Com cadastro de conceitos)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções utilizando a opção:
 - Adição de conceitos
 - Adicionar um novo conceito
 - Inserir uma descrição para o conceito
 - Inserir uma imagem para o conceito (opcional)
 - Salvar o caso, escolhendo um local para gravar o arquivo descritivo do caso.
- **Resultado esperado** – Cadastro com sucesso

2.6 - Cadastro dos casos (Com cadastro de conceito já existente)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – Conceito esperado já existente na base
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções utilizando a opção:
 - Adição de conceitos
 - Adicionar um novo conceito
 - Inserir uma descrição para o conceito
 - Inserir uma imagem para o conceito (opcional)
- **Resultado esperado** - Mostrar mensagem de conceito já existente

2.7 - Cadastro dos casos (Utilizando sugestão – funções encontradas)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – Funções semelhantes existentes na base
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa

- Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções utilizando a opção de sugestão:
 - Selecionar dentre as opções apresentadas as que podem ser relacionadas com a função da matriz em questão
 - Conceitos serão automaticamente preenchidos de acordo com as funções selecionadas
 - Salvar o caso, escolhendo um local para gravar o arquivo descritivo do caso.
- **Resultado esperado** – Cadastro com sucesso

2.8 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – Não existirem funções semelhantes na base
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções utilizando a opção de sugestão.
- **Resultado esperado** - Mostrar mensagem de funções não encontradas

2.9 - Cadastro dos casos (Utilizando sugestão – funções semelhantes não encontradas)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – nenhuma
- **Fluxo de ações esperado**
 - Inserir os dados iniciais do caso na tela de cadastro
 - Inserir as informações da função geral
 - Inserir os desdobramentos da função geral e para cada função:
 - Preencher os dados de classificação, até que o ícone no topo se altere para avisar que a função está completa
 - Marcar a caixa de seleção, caso esta função seja uma função elementar (que irá para a matriz morfológica)
 - Inserir os novos desdobramentos caso haja algum
 - Alterar para a aba da matriz morfológica
 - Preencher a fila de conceitos para cada uma das funções utilizando a opção de sugestão.
- **Resultado esperado** - Mostrar mensagem de funções não encontradas

2.10 - Cadastro dos casos (Utilizando edição de caso já existente)

- **Caminho** – Abrir aplicativo → Cadastro de Casos
- **Descrição da funcionalidade** – Inserir casos na base
- **Condições Prévias** – Ter casos cadastrados previamente
- **Fluxo de ações esperado**
 - Selecionar a opção de editar casos
 - Mostrar tela com todos os casos cadastrados e opção de busca
 - Selecionar caso da lista
- **Resultado esperado** – preencher caso selecionado com nome alterado (variação numérica) para edição

3 – Importação/Exportação de informações (importação)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos → importar
- **Descrição da funcionalidade** – Inserir casos na base via arquivo
- **Condições Prévias** – Ter um caso em arquivo
- **Fluxo de ações esperado**
 - Selecionar o arquivo a ser importado
- **Resultado esperado** – Caso importado mostrado na listagem dos casos, juntamente com sua decomposição funcional e matriz morfológica

3.1 – Importação/Exportação de informações (importação falha)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos → importar
- **Descrição da funcionalidade** – Inserir casos na base via arquivo
- **Condições Prévias** – Ter um caso em arquivo
- **Fluxo de ações esperado**
 - Selecionar o arquivo a ser importado
- **Resultado esperado** – mensagem de erro indicando arquivo inválido para importação e a razão

3.2 – Importação/Exportação de informações (exportação)

- **Caminho** – Abrir aplicativo → Informações → Buscar casos → exportar
- **Descrição da funcionalidade** – Inserir casos na base via arquivo
- **Condições Prévias** – Ter um caso cadastrado
- **Fluxo de ações esperado**
 - Selecionar o arquivo a ser exportado da listagem de casos
 - Selecionar o local aonde o mesmo deverá ser salvo
- **Resultado esperado** – Arquivo “.cbm” gerado no local escolhido com as informações do caso em questão.

Ao final desta etapa, o software deverá estar livre de erros de programação, deixando-o pronto para o uso nos testes de desempenho e aceitação.

Teste de Desempenho

No ambiente de desenvolvimento, serão feitos os testes de erro e mais um teste de desempenho com o objetivo de verificar o comportamento do sistema em relação à quantidade de casos cadastrados na base.

O teste de desempenho será efetuado após os testes de aceitação por depender diretamente dos dados gerados pelos alunos durante a sua execução.

Para o teste de performance, um projeto de um sistema mecatrônico foi separado e será utilizado da seguinte maneira:

- Será feito o seu cadastro com a base em seu estado atual (25 projetos cadastrados). Será utilizada a opção de sugestão para gerar os conceitos e verificar o quanto e se estes conseguem se aproximar do projeto real.
- Após o teste de aceitação, efetuados em ambiente acadêmico, os dados gerados por este teste serão coletados e adicionados na base atual para que a etapa acima seja repetida e verificado se houve melhoria dos conceitos sugeridos pelo sistema.

Os parâmetros utilizados para avaliação de desempenho serão:

- Qualidade da recuperação das funções (funções associadas corretamente x funções associadas incorretamente)
- Qualidade dos conceitos atribuídos (quantidade de conceitos aproveitáveis dentre os sugeridos)

Espera-se que ao final desta etapa seja possível compreender e validar o correto emprego dos conceitos do RBC na criação de uma ferramenta capaz de utilizar os casos do histórico para sugerir soluções cabíveis aos seus usuários.

Análise do uso (teste de aceitação)

O teste de aceitação será feito em uma sala de aula com X alunos, divididos em grupos de Y pessoas. O aplicativo será instalado previamente em todas as máquinas disponíveis para uso da classe. O projeto que será utilizado nos testes será o desenvolvido pelos grupos ao longo do semestre.

Ao início dos testes, será dada uma breve explicação sobre o uso do aplicativo e suas funcionalidades básicas. Os alunos serão solicitados a utilizá-la para fazer cada um, uma variação do seu projeto conceitual. Cada um dos alunos deverá utilizar o aplicativo ao menos uma vez. Será dado o tempo de uma aula para que os alunos efetuem o cadastro de seus projetos, após o qual cada um deverá receber o formulário de uso para preenchimento e posterior avaliação.

Todos os dados gerados para cada um dos projetos deverá ser enviado ao e-mail jmtstorres@gmail.com.

O questionário para avaliação do uso foi dividido de acordo com os requisitos, para verificação da competência do aplicativo em realizar cada um deles. As perguntas serão respondidas de acordo com uma escala (resposta em escala de 0-5 – 0 pouco, 5 muito), exceto quando especificado e são mostradas abaixo separadas em grupos, de acordo com o requisito avaliado (para a entrevista com os alunos, os itens serão inseridos em um formulário):

- Interface
 - Falhas são mostradas de forma eficiente
 - Funcionalidades desejadas são encontradas com facilidade
 - Os itens de ajuda são visíveis
 - Os itens de ajuda possuem informações relevantes
- Cadastrar novos casos
 - A montagem dos desdobramentos das funções é intuitiva
 - Os parâmetros que descrevem as funções são pertinentes
 - A montagem da matriz morfológica é intuitiva
- Editar casos já inseridos
 - A funcionalidade de edição dos casos facilita a geração de variações para o projeto

- Decomposição funcional que mostre de forma sucinta os desdobramentos de funções e permita rápida edição dos parâmetros das funções e seus desdobramentos
 - Interface familiar aos conceitos de projeto de produto conceitual
- Matriz morfológica que permita a inserção dos conceitos automaticamente, com a funcionalidade de sugestão
 - As funções foram relacionadas como semelhantes de forma correta
 - Os conceitos sugeridos foram úteis para o projeto
 - A ferramenta de sugestão foi útil ao projeto
 - O uso do software agilizou o projeto conceitual como um todo
 - O uso do software agilizou no desdobramento das funções
 - O uso do software agilizou a composição da matriz
- Matriz morfológica que permita a inserção dos conceitos manualmente, a partir de conceitos previamente armazenados
 - A seleção de conceitos é intuitiva
 - A seleção de conceitos é rápida
- Matriz morfológica que permita a inserção dos conceitos manualmente, a partir de novos conceitos cadastrados imediatamente
 - O cadastro de conceitos é intuitivo
 - O cadastro de conceitos é rápido
- Prover informações sobre os casos cadastrados de forma rápida e simples
 - Não se aplica
- Fornecer a solução de modo legível e permitir alterações prévias ao armazenamento de um novo caso
 - A interface de sugestão de conceitos é intuitiva
- Permitir a exportação da base para outros sistemas e importação de casos de outras bases
 - Dificuldade ao exportar os casos da base
 - Dificuldade ao importar os casos da base
- Arquitetura modular para permitir o uso das funções do motor em sistemas externos
 - Não se aplica
- Da ferramenta (geral)

- Indique um (ou mais) ponto(s) de melhora da ferramenta avaliados a partir do seu uso (discursiva)
- Qual sua dificuldade no uso do software
- O cadastro do caso ocorreu sem falhas (sim/não)
- O usuário conseguiu cadastrar todos os casos (sim/não)
- Foram necessárias mais de uma tentativa para cadastro do caso(sim/não)
- Caso tenha encontrado uma situação de erro que impediu o seu uso, descreva (discursiva)

Ao fim desta última etapa, espera-se validar a utilidade da ferramenta em ambiente de projeto (ainda que acadêmico) agilizando e facilitando a tarefa dos projetistas na etapa de projeto conceitual.

Considerações finais

Ao longo deste documento foram descritas todas as etapas planejadas para a validação da ferramenta. Espera-se que os dados aqui levantados possam auxiliar na pesquisa desenvolvida mostrando os pontos positivos e negativos da utilização destes conceitos no ambiente de projeto.

Formulário de Avaliação

Projeto:

Responda, somente se especificado em contrário, utilizando a escala (0 – pouco, 5 - muito)

Interface

Falhas são mostradas de forma eficiente

Funcionalidades desejadas são encontradas com facilidade

Os itens de ajuda são visíveis

Os itens de ajuda possuem informações relevantes

Cadastro de novos casos

A montagem dos desdobramentos das funções é intuitiva

Os parâmetros que descrevem as funções são pertinentes

A montagem da matriz morfológica é intuitiva

Edição de casos

A funcionalidade de edição dos casos facilita a geração de variações para o projeto

Decomposição funcional

Interface familiar aos conceitos de projeto de produto conceitual

O uso do software agilizou no desdobramento das funções

Matriz morfológica

As funções foram relacionadas como semelhantes de forma correta

Os conceitos sugeridos foram úteis para o projeto

A ferramenta de sugestão foi útil ao projeto

O uso do software agilizou a composição da matriz

O cadastro de conceitos é intuitivo

O cadastro de conceitos é rápido

A interface de sugestão de conceitos é intuitiva

Importação/Exportação de casos

Dificuldade ao exportar os casos da base

Dificuldade ao importar os casos da base

Da ferramenta (geral)

O uso do software agilizou o projeto conceitual como um todo

Qual sua maior dificuldade no uso do software

O cadastro do caso ocorreu sem falhas (sim/não)

O usuário conseguiu cadastrar todos os casos (sim/não)

Foram necessárias mais de uma tentativa para cadastro do caso(sim/não)

Indique um (ou mais) ponto(s) de melhora da ferramenta avaliados a partir do seu uso (discursiva)

Caso tenha encontrado uma situação de erro que impediu o seu uso, descreva o ocorrido (discursiva)