



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Máquina de Raciocínio Lógico para Tomada de Decisões Estratégicas em Robotica Educacional

Autor: Carolina Barros Ramalho
Orientador: (Prof^a. Dr^a. Milene Serrano)

Brasília, DF
2015



Carolina Barros Ramalho

Máquina de Raciocínio Lógico para Tomada de Decisões Estratégicas em Robotica Educacional

Monografia submetida ao curso de graduação
em (Engenharia de Software) da Universi-
dade de Brasília, como requisito parcial para
obtenção do Título de Bacharel em (Enge-
nharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: (Prof^a. Dr^a. Milene Serrano)

Coorientador: (Prof. Dr. Maurício Serrano)

Brasília, DF

2015

Carolina Barros Ramalho

Máquina de Raciocínio Lógico para Tomada de Decisões Estratégicas em Robotica Educacional/ Carolina Barros Ramalho. – Brasília, DF, 2015-

83 p. : il. (algumas color.) ; 30 cm.

Orientador: (Prof^a. Dr^a. Milene Serrano)

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. Palavra-chave01. 2. Palavra-chave02. I. (Prof^a. Dr^a. Milene Serrano). II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Máquina de Raciocínio Lógico para Tomada de Decisões Estratégicas em Robotica Educacional

CDU 02:141:005.6

Errata

Elemento opcional da ??, 4.2.1.2). **Caso não deseje uma errata, deixar todo este arquivo em branco.** Exemplo:

FERRIGNO, C. R. A. **Tratamento de neoplasias ósseas apendiculares com reimplantação de enxerto ósseo autólogo autoclavado associado ao plasma rico em plaquetas:** estudo crítico na cirurgia de preservação de membro em cães. 2011. 128 f. Tese (Livre-Docência) - Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo, 2011.

Folha	Linha	Onde se lê	Leia-se
1	10	auto-conclavo	autoconclavo

Carolina Barros Ramalho

Máquina de Raciocínio Lógico para Tomada de Decisões Estratégicas em Robotica Educacional

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 01 de junho de 2013:

(Prof^a. Dr^a. Milene Serrano)
Orientador

Titulação e Nome do Professor
Convidado 01
Convidado 1

Titulação e Nome do Professor
Convidado 02
Convidado 2

Brasília, DF
2015

*Este trabalho é dedicado à minha mãe, Maria das Graças Barros Ramalho,
à minha irmã, Maysa Ramalho Lima, que fizeram de mim tudo o que sou hoje, e ao meu
namorado, Wisley Rocha da Silva, por estar ao meu lado e me apoiar durante essa longa
e árdua jornada./*

Agradecimentos

A inclusão desta seção de agradecimentos é opcional, portanto, sua inclusão fica a critério do(s) autor(es), que caso deseje(em) fazê-lo deverá(ão) utilizar este espaço, seguindo a formatação de *espaço simples e fonte padrão do texto (arial ou times, tamanho 12 sem negritos, aspas ou itálico*.

Caso não deseje utilizar os agradecimentos, deixar toda este arquivo em branco.

A epígrafe é opcional. Caso não deseje uma, deixe todo este arquivo em
branco.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

O resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. O texto pode conter no mínimo 150 e no máximo 500 palavras, é aconselhável que sejam utilizadas 200 palavras. E não se separa o texto do resumo em parágrafos.

Palavras-chaves: latex. abntex. editoração de texto.

Abstract

This is the english abstract.

Key-words: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Sensores para robôs móveis (WOLF et al., 2009)	42
Figura 2 – Atuadores para robôs móveis (WOLF et al., 2009)	43
Figura 3 – Classificação das pesquisas bibliográficas (GIL, 2002)	58
Figura 4 – Modelagem do processo utilizado no TCC	61

Lista de tabelas

Tabela 1 – Cronograma do TCC 01	62
Tabela 2 – Cronograma do TCC 02	62

Lista de abreviaturas e siglas

Fig. Area of the i^{th} component

456 Isto é um número

123 Isto é outro número

lauro cesar este é o meu nome

Lista de símbolos

Γ	Letra grega Gama
Λ	Lambda
ζ	Letra grega minúscula zeta
\in	Pertence

Sumário

1	INTRODUÇÃO	29
1.1	Contextualização	29
1.2	Justificativa	30
1.3	Questão de Pesquisa	31
1.4	Objetivos	32
1.4.1	Objetivo geral	32
1.4.2	Objetivos específicos	32
1.5	Organização dos capítulos	32
2	REFERENCIAL TEÓRICO	35
2.1	Robótica educacional	35
2.1.1	Perspectiva histórica da robótica	35
2.1.2	Definição de robô e robótica	36
2.1.3	Robótica como instrumento educacional	36
2.1.3.1	Teorias de aprendizagens	37
2.1.3.1.1	Howard Gardner – Inteligências múltiplas	37
2.1.3.1.2	Philippe Perrenoud – Ensino por competências	38
2.1.3.1.3	Jean Piaget – Construtivismo	38
2.1.3.1.4	Seymour Papert – Construcionismo	39
2.1.3.2	Conceito da robótica educacional	39
2.1.3.3	Objetivos da robótica educacional	40
2.1.4	Fundamentos da robótica móvel	41
2.1.4.1	Controle reativo	43
2.1.4.2	Controle deliberativo	44
2.1.4.3	Controle hierárquico	44
2.1.4.4	Controle híbrido	45
2.2	Engenharia de Software	45
2.2.1	Algoritmo guloso	45
2.2.1.1	Características do algoritmo guloso	45
2.2.1.2	Elementos da estratégia gulosa	46
2.2.1.2.1	Propriedade da escolha gulosa	46
2.2.1.2.2	Subestrutura ótima	46
2.2.1.3	Fundamentos do algoritmo guloso	47
2.2.2	Programação dinâmica	47
2.2.2.1	<i>Memoization</i> vs Iteração sobre subproblemas	48

2.2.3	Paradigma lógico	48
2.2.3.1	Surgimento da programação lógica	48
2.2.3.2	PROLOG	49
2.2.3.2.1	Fundamentos do PROLOG	49
2.3	Trabalhos relacionados	50
2.4	Considerações finais	50
3	SUPORTE TECNOLÓGICO	51
3.1	Robótica Educacional	51
3.1.1	Robomind	51
3.1.2	Scratch	52
3.1.3	Arduino	52
3.1.4	Raspberry Pi	52
3.1.5	Kit Educacional Lego Mindstorms	53
3.2	Engenharia de Software	53
3.2.1	Ubuntu	53
3.2.2	SWIProlog	54
3.2.3	Bizagi Process Modeler	54
3.2.4	Git	54
3.2.5	Github	54
3.2.6	Waffle	54
3.2.7	LaTeX	55
3.2.8	TexMaker	55
3.3	Considerações finais	55
4	METODOLOGIA	57
4.1	Classificação da pesquisa	57
4.1.1	Quanto aos Objetivos Gerais	57
4.1.1.1	Pesquisa Exploratória	57
4.1.1.2	Pesquisa Descritiva	57
4.1.1.3	Pesquisa Explicativa	58
4.1.2	Quanto aos Procedimentos	58
4.1.2.1	Pesquisa Bibliográfica	58
4.1.2.2	Pesquisa Documental	59
4.1.2.3	Pesquisa Experimental	59
4.1.2.4	Pesquisa <i>Ex-post Facto</i>	59
4.1.2.5	Levantamento	59
4.1.2.6	Estudo de Caso	60
4.2	Planejamento da pesquisa	60
4.3	Modelagem da metodologia	60

4.4	Cronograma da pesquisa	62
4.5	Considerações finais	62
5	PROVA DE CONCEITO	63
5.1	Descrição da prova de conceito	63
5.2	Considerações finais	63
6	PROPOSTA	65
6.1	Descrição da proposta	65
6.2	Considerações finais	65
7	RESULTADOS OBTIDOS	67
8	CONCLUSÃO	69
	Referências	71
	APÊNDICES	73
	APÊNDICE A – PRIMEIRO APÊNDICE	75
	APÊNDICE B – SEGUNDO APÊNDICE	77
	ANEXOS	79
	ANEXO A – PRIMEIRO ANEXO	81
	ANEXO B – SEGUNDO ANEXO	83

1 Introdução

Este documento apresenta considerações gerais e preliminares relacionadas à redação de relatórios de Projeto de Graduação da Faculdade UnB Gama (FGA). São abordados os diferentes aspectos sobre a estrutura do trabalho, uso de programas de auxílio a edição, tiragem de cópias, encadernação, etc.

1.1 Contextualização

O termo robô surgiu em meados do século XX, derivado da palavra tcheca *robota*, que significa trabalhador forçado (ou escravo) (SILVA, 2009). Desde esta época, a propagação do conceito de robôs está acontecendo de forma acelerada, seja por filmes de ficção científica, ou documentários, ou desenhos animados. Os robôs saíram, de fato, da ficção científica em 1961, quando Joseph Engelberger desenvolveu o primeiro robô comercial, o UNIMATE, e desde então estão cada dia mais inseridos em meio a sociedade, seja como elevadores, caixas eletrônicos, robôs de entretenimento ou de chão de fábrica (MURPHY, 2000 apud SILVA, 2009). Segundo (SILVA, 2009), um robô deve ter, idealmente, os seguintes elementos:

- atuadores: são os meios utilizados para que o robô se locomova e/ou altere a forma de seu corpo. Exemplo: pernas, rodas, articulações, garras, dentre outros;
- sensores: são os meios utilizados pelo robô para medir e conhecer o ambiente, detectando objetos, calor ou luz, e convertendo essa informação em símbolos processados por computadores;
- computador: é o responsável por controlar o robô através de algoritmos nele implementados;
- equipamentos ou mecanismos: São ferramentas ou equipamentos mecânicos.

Um robô pode ser categorizado em um dos três grupos, atualmente conhecidos: manipuladores, móveis e híbridos. Os robôs manipuladores são fixos ao seu local de trabalho; os móveis se locomovem por meio dos atuadores, e os híbridos são um composto das duas categorias anteriores (RUSSEL; NORVIG, 2004 apud SILVA, 2009).

A robótica é uma ciência, em rápida ascensão, que envolve áreas do conhecimento como: microeletrônica, computação, engenharia mecânica, inteligência artificial, física, neurociência, entre outras. Portanto, estuda tecnologias associadas a projeto, fabricação, teoria e aplicação dos robôs.

Sendo a robótica uma área tão interdisciplinar, usá-la como instrumento de aprendizagem é um tanto quanto benéfico, pois ensina a criança e/ou jovem a trabalhar em equipe, desenvolver o raciocínio lógico através de problemas concretos, estimulando a leitura, exploração, investigação, criatividade e organização. Além de aprimorar a parte motora do indivíduo ao trabalhar com o hardware do robô, também é aprimorado o raciocínio lógico e a abstração ao programar o software do robô.

Em meados dos anos 60, Seymour Papert, reconhecido matemático, educador e pesquisador do MIT (Instituto de Tecnologia de Massachusetts), criou a linguagem de programação LOGO. Essa linguagem foi utilizada nos kits educacionais da LEGO, conferindo o início do sistema educacional LEGO-LOGO. Nesse sistema, as crianças têm a possibilidade de construir seus robôs protótipos com os blocos de montagem e outros recursos do kit educacional da LEGO bem como programar com a linguagem LOGO, gerando o comportamento desejado nesses protótipos.

Um dos kits de robótica mais populares criados pela LEGO é o Mindstorms, que combina um computador programável, NTX ou RCX dependendo da versão, com motores elétricos, engrenagens, peças de encaixe, polias, roscas, dentre outros. Este kit contém cerca de mil peças LEGO, incluindo o computador, o CD-ROM do software Mindstorms, um transmissor infravermelho para envio de programas para o robô, um guia do construtor, motores, sensores, rodas, pneus, conectores e outros. Para aprimorar o aprendizado, a LEGO disponibiliza, além do kit de peças para montagem dos robôs, um tapete de missões a serem realizadas pelos mesmos. Cada missão do tapete tem uma pontuação máxima, que poderá ser alcançada se a missão for completada dentro do tempo e sem penalidades. É possível completar 'n' missões dentro do tempo limite.

1.2 Justificativa

Retomando aspectos apontados na contextualização, seguem algumas preocupações intrínsecas desse contexto, com as quais procura-se justificar as necessidades: (i) de uma investigação mais detalhada quanto à literatura associada; (ii) da elicitação de soluções candidatas, apoiadas na Engenharia de Software, na Inteligência Artificial e no Projeto e Análise de Algoritmos, e (iii) da implementação de uma solução dentre as elicitadas.

O curso de Engenharia de Software da UnB/FGA oferece uma disciplina chamada Robótica Educacional. Dentre os objetivos dessa disciplina, tem-se a intenção de formar grupos de granduandos para competições em robótica, âmbito regional, nacional e internacional. Nesse contexto, são estabelecidos desafios aos grupos de alunos matriculados na disciplina. Esses desafios orientam-se pelas propostas de atividades da First LEGO League (KAMEN; KRISTIENSEN, 2010). Uma preocupação intrínseca dos grupos par-

ticipantes consiste em lidar com diferentes variáveis, em tempo de competição. No caso, destacam-se: tempo total estabelecido para conclusão das missões, localização atual do robô e pontuação das missões.

No intuito de colaborar na formação de uma equipe competitiva, têm-se como preocupações a serem investigadas nesse Trabalho de Conclusão de Curso, principalmente:

- O fato do tempo total para a conclusão das missões ser relativamente curto para que todas sejam realizadas, logo são necessários algoritmos que permitam a seleção dessas missões considerando o tempo como um fator impactante.
- Não apenas o tempo, mas para realizar essa seleção de forma apropriada, outras variáveis devem ser consideradas, tais como:
 - Localização atual do robô, pois dependendo da posição do robô em relação ao tapete o tempo de locomoção para realizar uma missão irá mudar.
 - Pontuação das missões, pois pode existir uma missão 'X' que gaste o mesmo tempo para ser realizada que uma missão 'Y', porém 'X' tem uma maior pontuação que 'Y', logo será mais valioso executar a missão 'X' do que a missão 'Y'.

Diante do exposto, acredita-se que a elaboração de uma base de conhecimento orientada ao Paradigma Lógico (TUCKER; NOONAN, 2009, chap. 15), especificamente implementada com base em algoritmos da Inteligência Artificial, conferirá ao robô a capacidade de gerar um roteiro de missões, sendo esse o de maior pontuação possível de ser realizada dentro do tempo restante.

Pretende-se, com tal esforço, agregar valor na formação de uma equipe competitiva em robótica, disponibilizando uma máquina de raciocínio apoiada em algoritmos avançados e instigando os membros da equipe a refinarem essa máquina de forma evolutiva e continuamente.

1.3 Questão de Pesquisa

Este TCC buscará responder ao seguinte questionamento: É possível implementar uma inteligência artificial no robô LEGO Mindstorms, i.e. uma máquina de raciocínio lógico, de tal forma que dada uma posição (X_0 , Y_0) e um tempo restante, esse robô consiga gerar um roteiro de missões a serem executadas para alcançar a maior pontuação possível?

1.4 Objetivos

1.4.1 Objetivo geral

Prover uma máquina de raciocínio lógico, considerando um algoritmo específico para tomada de decisões, com a qual o robô deverá ser capaz de selecionar as missões, compondo um roteiro, de forma que a pontuação seja maximizada em relação ao tempo disponível, previamente estabelecido.

1.4.2 Objetivos específicos

Com base nas colocações apresentadas nos tópicos anteriores, são objetivos específicos deste TCC:

- Investigar algoritmos candidatos à solução da questão de pesquisa considerando, em um primeiro escopo dessa atividade, o estudo de algoritmos gulosos, programação dinâmica e Paradigma Lógico;
- Implementar uma solução que agregue ao robô a capacidade de gerar roteiros de missões de máxima pontuação a partir de um ponto inicial e um tempo restante.
- Estudar o funcionamento do kit educacional LEGO Mindstorms visando a realização de uma pesquisa exploratória/experimental que permita a verificação quanto a pertinência da solução estabelecida para o contexto investigado.
- Estabelecer uma metodologia de desenvolvimento, orientada às boas práticas da Engenharia de Software, no intuito de conduzir o processo investigativo, a implementação da solução bem como a verificação dessa solução no contexto acordado.

1.5 Organização dos capítulos

Este trabalho de conclusão de curso está organizado em sete capítulos, sendo o primeiro dedicado à Introdução. Os demais são brevemente descritos a seguir:

- Referencial Teórico: explana sobre a evolução da robótica educacional, principais conceitos do Paradigma Lógico e da programação dinâmica.
- Suporte Tecnológico: descreve as tecnologias utilizadas no desenvolvimento tanto da máquina de raciocínio quanto da prova de conceito, apresentando ferramentas, *plugins* e *kits* utilizados.
- Metodologia: aborda, dentre outros detalhes, o fluxo das atividades, o cronograma bem como modalidades de pesquisa e desenvolvimento.

-
- Proposta: descreve em detalhes a máquina de raciocínio desde a arquitetura até a implementação do algoritmo.
 - Resultados Iniciais: relata o status atual do projeto, apontando as divergências e convergências entre os resultados esperados e obtidos até o momento.
 - Considerações Finais: efetua uma breve reflexão sobre o trabalho e a lista atividades previstas para a continuação do mesmo.

2 Referencial teórico

2.1 Robótica educacional

2.1.1 Perspectiva histórica da robótica

Os robôs são máquinas que podem substituir o homem na execução das tarefas, e consequentemente, destinados a melhorar a produção e a qualidade de vida (ANTONIO et al., 1997 apud SANTOS, 2002). Desde os primórdios, o homem se sente atraído por esse mundo das máquinas, como os antigos Egípcios que adicionaram braços mecânicos às estátuas de seus deuses para serem operados por sacerdotes; e os Gregos que construíram estátuas movidas hidraulicamente; e os Chineses que entre os séculos XVIII e XIX construíram bonecas que transportavam chá (SANTOS, 2002). Os autômatos projetados pelos Gregos tinham uma função mais estética e contemplativa, já os projetados pelos Árabes tinham funcionalidade, deixando a par preocupações estéticas e de entretenimento (SANTOS, 2002). É perceptível que os homens, há muitos séculos, utilizam os robôs, mas sem o conhecimento da palavra, que foi divulgada a primeira vez em 1921, pelo checoslovaco Karel Capek, no seu romance *Rossum's Universal Robots*, onde descreveu os robôs como máquinas com braços trabalhando duas vezes mais que os humanos, de forma incansável, eficiente e obediente, que se tornariam malévolos e dominariam o mundo (SANTOS, 2002). Porém em 1950, Isaac Asimov defendeu, em sua obra literária *I Robot*, que a construção de robôs seguiria uma linha positiva e benéfica (apud Nof 1999), concebendo-os como autômatos de aparência humana mas desprovidos de sentimentos (SANTOS, 2002). Asimov, em sua obra, fala sobre a inteligência dos robôs de acordo com as seguintes leis da robótica (SANTOS, 2002):

- 1ª Lei: Um robô não pode prejudicar um ser humano, ou quando inativo, deixar um ser humano exposto ao perigo;
- 2ª Lei: Um robô deve obedecer às ordens dadas pelo ser humano, exceto se tais ordens estiverem em contradição com a 1ª Lei.
- 3ª Lei: Um robô deve proteger a sua própria existência desde que essa proteção não entre em conflito com a 1ª e a 2ª Lei.

Em 1985, Asimov acrescentou uma 4ª Lei que diz que "o robô não pode causar mal à humanidade ou por falta de ação permitir que a humanidade sofra perigos". Dessa forma, Asimov popularizou o termo robô e idealizou as três leis fundamentais da robótica (SILVA et al., 2008).

2.1.2 Definição de robô e robótica

A Divisão Internacional de Robótica da Sociedade de Engenharia de Manufatura define um robô como sendo um manipulador reprogramável, multifunções, utilizado para deslocar outros materiais ou objetos específicos através da programação de movimentos (SILVA et al., 2008). Um computador é manipulado pelo homem, logo não é considerado um robô pois este é um mecanismo inteligente que funciona de forma autônoma (CURCIO, 2008). (PIO; CASTRO; JÚNIOR, 2006) definem a robótica como:

a ligação inteligente entre a percepção e a ação. Trabalhar em Robótica significa estudar, projetar e implementar sistemas ou dispositivos que, com a utilização de percepção e de certo grau de “inteligência”, sejam úteis na realização de uma determinada tarefa, pré-definida ou não, que envolva interação física entre o sistema (ou dispositivo) e o meio onde a tarefa está sendo realizada.

Porém, uma definição mais formal para o termo robótica seria "uma ciência da engenharia aplicada, tida como uma combinação da tecnologia de máquinas operatrizes e ciência da computação", assim a robótica envolve várias disciplinas como engenharia mecânica, elétrica, inteligência artificial e utiliza o robô como principal instrumento (CURCIO, 2008). Os robôs podem ser categorizados em primeira, segunda e terceira geração, sendo progressivamente mais inteligentes (ANTONIO et al., 1997 apud SANTOS, 2002).

- Primeira geração: A única função inteligente desses robôs é a aprendizagem de uma sequência de ações de manipulação coordenadas por um operador humano usando uma unidade de comando. Os robôs dessa geração repetem sistematicamente as tarefas e ignoram possíveis alterações no meio externo, causando restrições no seu uso, como posicionamento no espaço, relacionamento com outras máquinas e a segurança das pessoas que ficam próximas ao robô.
- Segunda geração: esses robôs tiveram a adição de um processador à sua configuração, possibilitando a adaptabilidade ao ambiente, mesmo que de forma mínima, com utilização de sensores para auxiliar na localização, detecção de esforços e adaptação de seus movimentos às informações coletadas. Geralmente a área de atuação destes robôs estão ligados à manufatura autômata.
- Terceira geração: é uma versão mais recente dos robôs, onde é incorporado à sua configuração processadores múltiplos em que, cada operação em sincronia desempenha uma tarefa diferente, tornando um mesmo robô multitarefas.

2.1.3 Robótica como instrumento educacional

A robótica educacional é uma forma de inserir a tecnologia no meio educativo como forma de aprendizado, oferecendo aos alunos uma oportunidade de vivenciarem experiências semelhantes às que vivem na vida real.

2.1.3.1 Teorias de aprendizagens

2.1.3.1.1 Howard Gardner – Inteligências múltiplas

Howard Gardner, professor e psicólogo da *Harvard Graduate School of Education*, dedicou-se, desde a década de 80, ao estudo das capacidades simbólicas em crianças com QI normal e alto, desenvolvendo o que ele chamou de "inteligências múltiplas" que pode ser vista como tendo três princípios fundamentais:

- A inteligência não é algo que pode ser vista como tendo múltiplas habilidades, existem múltiplas inteligências, cada uma com sua habilidade e distintas entre si.
- As inteligências são interdependentes, logo se é avaliada a competência de uma pessoa em uma área esta avaliação não servirá para outras áreas.
- Apesar de serem interdependentes as inteligências podem sim trabalhar em conjunto, caso contrário nada seria feito e nenhum problema seria resolvido.

Gardner identificou essas inteligências distintas e classificou da seguinte forma:

- Inteligência linguística: é relacionada a habilidade da pessoa em produzir a linguagem escrita e falada.
- Inteligência lógico-matemática: é relacionada a habilidade para explorar relações, categorias e padrões, através da manipulação de objetos ou símbolos, de resolver problemas através do raciocínio.
- Inteligência musical: é relacionada, como o próprio nome já diz, no reconhecimento da estrutura musical, sensibilidade dos sons, criação de melodias e ritmos, percepção das qualidades dos tons e habilidade para tocar instrumentos.
- Inteligência espacial: é relacionada a habilidade do indivíduo de visualizar um objeto e ter uma percepção acurada de diferentes ângulos, relações de espaço, dentre outros.
- Inteligência cinestésica: é representada pela capacidade de manipular objetos habilmente e controlar os movimentos do corpo.
- Inteligência interpessoal: é relacionada a capacidade de diferenciar e dar uma resposta aos estados de humor, temperamentos, desejos e motivações das outras pessoas.
- Inteligência intrapessoal: é relacionada a habilidade para ter acesso aos próprios sentimentos, aos estados interiores do ser, e saber utilizá-los na solução de problemas pessoais.

Segundo Gardner, a escola valoriza mais a inteligência linguística e a lógico-matemática, e o uso do computador pode colaborar para o amadurecimento das outras inteligências, por ser uma ferramenta adaptável as mais diversas formas de uso.

2.1.3.1.2 Philippe Perrenoud – Ensino por competências

Philippe Perrenoud é um professor e sociólogo suíço, referência para diversos educadores com suas idéias pioneiras e vanguardistas sobre profissionalização de professores e avaliação de alunos. Perrenoud afirma que "a trilogia das habilidades de ler, escrever, contar, que fundou a escolaridade obrigatória no século XIX não está mais a altura das exigências da nossa época". Assim, Perrenoud propôs o ensino por competências que considera que os saberes são ferramentas para a ação, que se aprende a usá-los, como as outras coisas, não valorizando o aspecto do acúmulo de saberes. O Centro de Referência Educacional elencou dez competências que devem ser trabalhadas pela escola:

1. Respeitar as identidades e as diferenças;
2. Utilizar-se das linguagens como meio de expressão, comunicação e informação;
3. Inter-relacionar pensamentos, idéias e conceitos;
4. Desenvolver o pensamento crítico e flexível e a autonomia intelectual;
5. Adquirir, avaliar e transmitir informações;
6. Compreender os princípios das tecnologias e suas relações integradoras;
7. Entender e ampliar fundamentos científicos e tecnológicos;
8. Desenvolver a criatividade;
9. Saber conviver em grupo;
10. Aprender a aprender.

2.1.3.1.3 Jean Piaget – Construtivismo

Jean Piaget, suíço, estudou como o aprendiz passa de um estado de menor conhecimento para outro de maior conhecimento, e desenvolveu a teoria do construtivismo que enfoca o conhecimento científico na perspectiva do indivíduo que aprende. Nessa teoria o sujeito é um ser ativo que estabelece relação de troca com o meio-objeto, relações essas que devem ser vivenciadas e significativas, assim, o indivíduo incorpora novas informações, que passa a tornar parte do conhecimento.

O construtivismo defende que o conhecimento não é uma cópia da realidade, mas sim uma construção do ser humano em consequência da sua interação com o ambiente e resultado de suas disposições internas.

Piaget classifica os períodos de inteligência em estágios, organizados do nascimento até a fase adulta. Esses estágios indicam saltos bruscos nas capacidades do indivíduo pois as capacidades cognitivas sofrem uma forte reestruturação.

2.1.3.1.4 Seymour Papert – Construcionismo

Seymour Papert, psicólogo do Laboratório de Inteligência Artificial do MIT, adaptou os princípios do construtivismo definido por Piaget e criou a teoria construcionista, que considera o computador como uma ferramenta para a construção do conhecimento e desenvolvimento do aluno. Duas idéias principais sobre a construção do conhecimento fazem com que o construcionismo se diferencie do construtivismo:

1. O aprendiz é quem constrói o seu conhecimento, através das coisas que faz;
2. O aprendiz constrói algo do seu interesse e que o motiva.

Segundo Papert, a criança aprende de forma mais eficaz quando, por elas mesmas, atingem o conhecimento específico de que precisam. O construcionismo tem como meta ensinar de forma a produzir o maior conhecimento possível com o mínimo de ensino.

2.1.3.2 Conceito da robótica educacional

maisonnette2002utilizacao define robótica como sendo

o controle de mecanismos eletro-eletrônicos através de um computador, transformando-o em uma máquina capaz de interagir com o meio ambiente e executar ações decididas por um programa criado pelo programador a partir destas interações. Podemos exemplificar o uso da robótica em diversas áreas de conhecimento.

e diz que a robótica educacional é a aplicação dessa tecnologia como mais uma ferramenta de ensino para que os alunos vivenciem de forma real o que estudam na prática, tendo a oportunidade e propor e solucionar problemas difíceis ao invés de apenas observar a solução. Através da robótica educacional os alunos podem explorar idéias e descobrir novos caminhos na aplicação de conceitos que aprende em sala de aula, assim adquirem a capacidade de produzir hipóteses, averiguar soluções, estabelecer relações e inferir conclusões (BENITTI et al., 2009). Além de benéfico ao aluno, a robótica educacional colabora com o professor no ensino de muitos conceitos teóricos de difícil compreensão, motivando o aluno a observar, abstrair e inventar. O conhecimento adquirido através de observação, abstração e do esforço do aluno tem muito mais significado para ele, e se adapta melhor às suas estruturas mentais (ZILLI et al., 2004).

2.1.3.3 Objetivos da robótica educacional

Segundo (ZILLI et al., 2004), além de propiciar o conhecimento da tecnologia atual a robótica educacional pode desenvolver as seguintes competências:

- raciocínio lógico;
- habilidades manuais e estéticas;
- relações inter e intrapessoais;
- utilização dos conceitos aprendidos em diversas áreas do conhecimento para o desenvolvimento de projetos;
- investigação e compreensão;
- representação e comunicação;
- trabalho com pesquisa;
- resolução de problemas por meio de tentativa e erro;
- aplicação das teorias à atividades concretas;
- utilização da criatividade em diversas situações;
- capacidade crítica.

(ZILLI et al., 2004) apresenta ainda uma classificação dos objetivos:

- Objetivos Gerais
 - Contruir maquetes que utilizem lâmpadas, motores e sensores;
 - Trabalhar conceitos de desenho, física, álgebra e geometria;
 - Conhecer e aplicar princípios de eletrônica digital;
 - Construir ou adaptar elementos dinâmicos como engrenagens, redutores de velocidade de motores, entre outros.
- Objetivos Psicomotores
 - Desenvolver a motricidade fina;
 - Proporcionar a formação de habilidades manuais;
 - Desenvolver a concentração e observação;
 - Motivar a precisão de seus projetos.
- Objetivos Cognitivos

- Estimular a aplicação das teorias formuladas à atividades concretas;
 - Desenvolver a criatividade dos alunos;
 - Analisar e entender o funcionamento dos mais diversos mecanismos físicos;
 - Ser capaz de organizar suas idéias a partir de uma lógica mais sofisticada de pensamento;
 - Selecionar elementos que melhor se adequam à resolução dos projetos;
 - Reforçar conceitos de matemática e geometria;
 - Desenvolver noções de proporcionalidade;
 - Desenvolver noções topológicas;
 - Reforçar a aprendizagem da linguagem Logo;
 - Introduzir conceitos de robótica;
 - Levar à descoberta de conceitos de física de forma intuitiva;
 - Utilizar conceitos aprendidos em outras áreas do conhecimento para o desenvolvimento de um projeto;
 - Proporcionar a curiosidade pela investigação levando ao desenvolvimento intelectual do aluno.
- Objetivos Afetivos
 - Promover atividades que geram a cooperação em trabalhos em grupo;
 - Estimular o crescimento individual através da troca de projetos e idéias;
 - Garantir que o aluno se sinta interessado em participar de discussões e trabalhos em grupo;
 - Desenvolver o senso de responsabilidade;
 - Despertar a curiosidade;
 - Motivar o trabalho de pesquisa;
 - Desenvolver a autoconfiança e a auto-estima;
 - Possibilitar a resolução de problemas por meio de erros e acertos.

2.1.4 Fundamentos da robótica móvel

O foco da robótica vem se modificando ao longo dos anos. Tempos atrás, os braços mecânicos, também chamados de manipuladores, tinham um maior destaque junto a mídia e a sociedade em geral (WOLF et al., 2009) sendo até, um dos principais fatores para a Revolução Industrial. Nos tempos atuais, o destaque é a robótica móvel, que trata de máquinas capazes de se movimentar de forma independente, podendo ser terrestre, aquático, voador ou até espacial (??, ?? apud ??, ??), movidos por rodas, esteiras, patas, hélices e dentre outros. Segundo (WOLF et al., 2009) os Robôs Móveis Autônomos (RMA, possuem como características fundamentais

as capacidades de locomoção e de operação de modo semi ou completamente autônomo. Também deve ser considerado que maiores níveis de autonomia serão alcançados somente à medida que o robô passe a integrar outros aspectos considerados da maior importância, como: *capacidade de percepção* (sensores que conseguem “ler” o ambiente onde ele atua), *capacidade de agir* (atuadores e motores capazes de produzir ações, tais como o deslocamento do robô no ambiente), *robustez e inteligência* (capacidade de lidar com as mais diversas situações, de modo a resolver e executar tarefas por mais complexas que sejam).

O termo Controle Robótico Inteligente é referente ao uso de técnicas de planejamento e controle para a navegação e operação autônoma dos robôs, que permitem que os RMAs tenham a capacidade de executar as mais diversas e complexas tarefas. Esse controle inteligente é possível devido ao uso de diversos sensores e atuadores que combinados conferem ao robô a possibilidade de planejar e realizar o acionamento de seus dispositivos de modo a executar a ação desejada. Sensores e atuadores para robôs móveis são listados a seguir.

Sensor	Principal Função	Exemplos
De Posição e Orientação	Determinar a posição absoluta ou direção de orientação do robô	GPS (Sistema de Posicionamento Global)
		Bússola [Compass]
		Inclinômetro
		Triangulação usando marcas (Beacons)
De Obstáculos	Determinar a distância até um objeto ou obstáculo	Sensor Infra-Vermelho (IR - Infrared)
		Ultrassom (Sonar)
		Radar
		Sensor Laser (Laser rangefinder)
De Contato	Determinar o contato com um objeto ou posição de contato com marcação	Sistemas de Visão Estéreo (Stereo Vision)
		Sensores de Contato (Bumpers, Switches)
		Antenas e "bigodes" (Animal whiskers)
		Marcações (barreiras óticas e magnéticas)
De Deslocamento e Velocidade	Medir o deslocamento do robô Medidas relativas da posição e orientação do robô	Inercial (Giroscópio, Acelerômetros)
		Odômetro (Encoders: Optical, Brush)
		Potenciômetros (Angular)
		Sensores baseados em Visão
Para Comunicação	Envio e recepção de dados e sinais externos (troca de informação)	Sistemas de Visão e Sensores Óticos
		Sistemas de Comunicação (RF)
Outros tipos	Sensores magnéticos, indutivos, capacitivos, reflexivos Sensores de temperatura, carga (bateria), pressão e força, etc. Detectores: detector de movimento, de marcações, de gás/odores	

Figura 1 – Sensores para robôs móveis (WOLF et al., 2009)

Atuador	Principal Tipo/Função	Exemplos
Base Fixa	Braço robótico com base fixa	Robôs industriais PUMA
Base Móvel: Rodas	2 Rodas independentes (diferencial)	Robôs Khepera e Pioneer 3-DX
	3 Rodas (triciclo, omni-directionais)	Robô BrainStem PPRK
	4 Rodas (veículos robóticos - ackermann)	Stanley - Stanford (Darpa Challenge)
Base Móvel: Esteira	Esteira (Slip/Skid locomotion - tracks)	Tanques e veículos militares
Base Móvel: Juntas e Articulações	Bípedes	Robôs Humanóides
	4 Patas (quadpods)	Robôs Sony Aibo, BigDog
	6 Patas (hexapods)	Robôs Inseto (Lynxmotion Hexapods)
Base Móvel: Propulsão Hélices ou Turbinas	Veículos aéreos com hélices	Aviões, Helicópteros e Dirigíveis
	Veículos aquáticos com hélices	Barcos autônomos
	Veículos sub-aquáticos	Submarinos autônomos
Outros tipos	Braços manipuladores com base móvel	Garras (Grippers) embarcadas
	Garras com ou sem feed-back sensorial	Mão robótica
	Mecanismos de disparo	Disparo do chute (futebol de robôs)

Figura 2 – Atuadores para robôs móveis (WOLF et al., 2009)

Os sensores conferem ao robô uma visão parcial, incompleta e sujeita a erros, sendo tarefa do controle inteligente adquirir, unificar e tratar as informações providas pelos sensores. Os comandos dos atuadores também são imprecisos, pois estes estão sujeitos a erros de posicionamento do robô e acionamento dos motores além das forças externas, como fricção, gravidade, colisão com obstáculos, derrapagem de rodas, dentre outros, logo cabe ao controle inteligente prover técnicas que permitam contornar estes cenários, corrigindo os erros de modo que as tarefas do robô sejam executadas corretamente (WOLF et al., 2009).

Um projeto de sistema de controle de robô deve levar em conta uma série de quesitos como: i) Tipo de tarefa do robô; ii) Tipo e precisão dos sensores embarcados; iii) Tipo e precisão dos atuadores. Além de adotar uma arquitetura específica de controle que seja capaz de realizar algumas, ou no melhor dos casos todas, as seguintes tarefas: i) Fusão de sensores; ii) Desviar de obstáculos; iii) Auto-localização; iv) Mapeamento do ambiente v) Planejamento de trajetórias; vi) Planejamento de ações; vii) Navegação robótica e viii) Interação e comunicação. O detalhamento de cada tópico pode ser conferido na íntegra na obra de (WOLF et al., 2009). Para que essas tarefas sejam planejadas e executadas gerenciando todos os dispositivos embarcados no robô é necessário fazer o uso das arquiteturas computacionais de controle de robôs móveis autômatos. Essas arquiteturas computacionais são relativas a percepção, raciocínio/decisão e ação em relação ao ambiente (WOLF et al., 2009). Dentre os diversos tipos de arquiteturas computacionais as mais conhecidas e reconhecidas são descritas a seguir.

2.1.4.1 Controle reativo

O controle reativo consiste em um sistema de reação sensorial-motora que considera apenas as leituras sensoriais realizadas no presente para fins de tomada de decisão e geração de comandos de ação (WOLF et al., 2009). Esse tipo de arquitetura computa-

cional tem por objetivo possibilitar a implementação de sistemas de controle com rápida resposta a um grande número de ocorrências ou situações de ambiente, permitindo que os robôs atuem em ambientes dinâmicos, devido a simplicidade no tratamento das informações sensoriais e a forma direta pela qual a percepção está associada com a ação (GRASSI, 2006 apud INDUSTRIAIS, 2009).

(WOLF et al., 2009) defende que o sistema reativo é muito útil para implementar comportamentos.

Um sistema reativo é bastante útil para implementar comportamentos elementares como: desviar de obstáculos (*avoid collision behaviour*: reage a presença de um obstáculo), seguir um objeto (*wall-following behaviour*: acompanhar um elemento guia) e seguir uma fonte luminosa (*phototaxis behaviour*: mover em direção a uma fonte de luz).

2.1.4.2 Controle deliberativo

A arquitetura deliberativa consiste em aplicar um mecanismo de planejamento e tomada de decisão, que estabeleça um plano prévio de execução de uma sequência de ações, baseado no modelo interno de conhecimento do mundo que o robô possui. Esse conhecimento prévio do mundo confere ao robô a possibilidade de otimizar seu desempenho em relação ao modelo interno.

O robô pode ter seu modelo interno representado de duas formas diferentes, simbólico e geométrico. O modelo simbólico é baseado em lógica, utilizando, por muitas vezes, a inteligência artificial. Já o modelo geométrico o mundo é representado espacialmente, por espaços livres e com obstáculos. (INDUSTRIAIS, 2009)

Tendo em vista que o robô possui esse modelo interno de mundo previamente estabelecido, ele possui limitações quando colocado em um mundo dinâmico, pois será necessário reconhecer o novo obstáculo, acrescentar ao seu modelo e reestabelecer o plano de execução (INDUSTRIAIS, 2009). Devido a essa limitação (WOLF et al., 2009) diz que o melhor seria a combinação dos modelos reativos e deliberativos, assim o robô teria capacidade de reação e de planejamento de tarefas complexas.

2.1.4.3 Controle hierárquico

O controle hierárquico consiste na combinação de múltiplos módulos de controle (reativo e/ou deliberativo) dispostas em forma hierárquica, podendo ser com decomposição vertical ou horizontal pra definir um esquema de prioridades em relação as múltiplas camadas do sistema (WOLF et al., 2009).

O controle hierárquico com decomposição vertical possui os módulos de controle reativo dispostos de forma vertical, mantendo a prioridade de baixo para cima, sendo a camada mais baixa a mais prioritária. Um exemplo de uso desse modelo é quando a ação

de desviar de um obstáculo, que está mais baixa, tem prioridade à ação de seguir em frente, que está mais acima.

O controle hierárquico com decomposição horizontal é adotada para controle de módulos deliberativos, implementando um "pipeline" que decompõe as funções em camadas executadas em cascata, formando uma linha de ação horizontal. Um exemplo desse modelo é o tipo Sense-Model-Plan-Act (SMPA) que possui um longo ciclo de resposta ao estímulo de entrada devido as suas quatro camadas horizontais que serão executadas em cascata para cada estímulo.

2.1.4.4 Controle híbrido

A arquitetura de controle híbrido nada mais é do que um controle reativo sendo planejado por um controle deliberativo, isto é, inicialmente é realizado um plano de execução pelo módulo deliberativo que, depois de pronto, é executado por um módulo reativo de maior prioridade que pode intervir nesse plano quando encontra algum obstáculo em meio à execução das ações. Nesse controle pode-se ainda ter um módulo de auto-localização que monitora constantemente a posição atual do robô e a corrige, se necessário.

Este tipo de arquitetura é uma das mais sofisticadas e adotadas na implementação dos sistemas de controle dos robôs móveis autônomos modernos, devido a sua eficiência em atingir os objetivos do robô.

2.2 Engenharia de Software

2.2.1 Algoritmo guloso

Os algoritmos gulosos decidem tomando como base apenas a informação disponível no dado momento, sem levar em conta as consequências da decisão, nunca reconsiderando uma decisão já tomada. Dessa forma, o algoritmo guloso toma uma decisão ótima naquele momento (local), e espera que essa decisão seja também a ótima global (ROCHA, 2004).

2.2.1.1 Características do algoritmo guloso

Segundo (ROCHA, 2004), os problemas e os algoritmos gulosos que os resolvem são caracterizados pelos itens a seguir.

- Para que exista uma solução ótima de um problema deve existir um conjunto de candidatos;
- Ao executar um algoritmo guloso dois conjuntos são criados: um que contém os elementos que foram analisados e rejeitados e outro os elementos que foram analisados e escolhidos;

- Um algoritmo guloso contém quatro funções:
 - a primeira avalia se o conjunto candidato produz uma solução para o problema;
 - a segunda verifica a viabilidade do conjunto de candidatos, ou seja, se é possível acrescentar mais candidatos a esse conjunto;
 - a terceira função, chamada de função seleção, avalia qual dos candidatos restantes é o melhor para ser acrescentado no conjunto;
 - a quarta e última função, chamada função objetivo, retorna o valor da solução encontrada.

É possível dizer que um algoritmo guloso trabalha da seguinte forma: inicialmente contém um conjunto S que está vazio, ou seja, nenhum candidato foi escolhido. A cada passo a função seleção é utilizada para determinar qual o melhor candidato, se o elemento analisado não é viável, ignora-se o termo que está sendo avaliado no momento, se for viável adiciona o elemento ao conjunto S . O elemento avaliado, sendo ele aceito ou rejeitado, será avaliado apenas uma vez, não havendo reconsiderações. A cada vez que o conjunto de candidatos escolhidos (S) é ampliado é verificado se a solução do problema foi obtida (ROCHA, 2004).

2.2.1.2 Elementos da estratégia gulosa

Como exposto, a estratégia utilizada pelo algoritmo guloso para encontrar a solução ótima do problema em questão é em cada passo escolher a solução que parece ser ótima naquele momento. Nem sempre uma estratégia gulosa é capaz de chegar à solução de um problema, mesmo esta existindo, mas existem duas características que podem indicar se o problema pode ou não ser resolvido através de uma estratégia gulosa.

2.2.1.2.1 Propriedade da escolha gulosa

Segundo a propriedade da escolha gulosa, uma solução globalmente ótima pode ser alcançada escolhendo uma solução localmente ótima. Para se provar que uma escolha localmente ótima em cada um dos passos irá levar a uma solução ótima é necessário examinar uma solução ótima global para algum subproblema e depois mostrar que ela pode ser modificada em uma solução gulosa, isto irá resultar em um subproblema menor, mas similar.

2.2.1.2.2 Subestrutura ótima

Esta característica também é importante quando se utiliza a programação dinâmica, vide seção 2.2.2. Um problema possui uma subestrutura ótima quando uma solução ótima para o problema contém, dentro dela, soluções ótimas para os subproblemas.

2.2.1.3 Fundamentos do algoritmo guloso

A teoria de algoritmos gulosos envolve o conceito de matróide que por definição é um par (E, I) onde E é um conjunto finito e $I \subseteq 2^E$, que satisfaz os seguintes axiomas (CASTALONGA; LEMOS et al., 2007):

- (i) $\emptyset \in I$;
- (ii) Se $X \subseteq Y$ e $Y \in I$, então $X \in I$;
- (iii) Se $X, Y \in I$ e $|X| < |Y|$ então existe $e \in Y - X$ tal que $X \cup e \in I$.

Esse tipo de estrutura não cobre todo tipo de problema que pode ser resolvido pelo algoritmo guloso, mas cobre muitos casos de interesse prático.

Lema: todo conjunto independente maximal em um matróide tem o mesmo tamanho. **Prova:** Por contradição. Se existir dois conjuntos independentes maximais A e B e $|A| < |B|$, deve existir um elemento $x \in (B - A)$ tal que $(A + x) \in I$. Isso contradiz o fato de que A é um conjunto maximal.

2.2.2 Programação dinâmica

A programação dinâmica assim como os métodos de "dividir para conquistar" combina as soluções dos subproblemas para resolver um problema. Esses métodos particionam os problemas em subproblemas de forma que resolvendo os subproblemas, recursivamente, resolve-se o problema, já a programação dinâmica aplica-se quando os subproblemas se particionam em subsubproblemas. Um algoritmo comum resolve repetidamente essas partições, trabalhando mais que o necessário, enquanto que o algoritmo dinâmico resolve cada subsubproblema apenas uma vez, salvando o resultado em uma tabela, evitando calcular repetidas vezes o mesmo problema (CORMEN, 2009). Um problema pode ter várias soluções possíveis, cada uma com um valor, e a solução com o melhor valor (maior ou menor) é dita a solução ótima para o problema. Segundo (CORMEN, 2009) para se desenvolver uma solução de programação dinâmica segue-se quatro passos:

- Caracterizar a estrutura de uma solução ótima;
- Recursivamente definir o valor de uma solução ótima;
- Calcular o valor de uma solução ideal, normalmente de forma ascendente;
- Construir uma solução ótima a partir de informações computadas.

Os três primeiros itens formam a base de uma solução dinâmica. Mas, quando utilizar a programação dinâmica para resolver um problema? O primeiro passo para resolver

um problema de otimização por programação dinâmica é caracterizar uma subestrutura ótima, sempre que um problema exibe uma subestrutura ótima. Para caracterizar o espaço de subproblemas, diz para manter o espaço o mais simples possível e, em seguida, expandi-lo, se necessário. Uma subestrutura ótima varia entre os domínios de problema de duas formas:

- (i) quantos subproblemas uma solução ideal utiliza para o problema, e
- (ii) quantas escolhas existem para determinar quais subproblemas usar em uma solução ótima.

Ou seja, pode ser que um problema tenha apenas um subproblema (de tamanho $n - i$), mas deve-se considerar n escolhas para i , afim de determinar quais escolhas produzem uma solução ótima.

2.2.2.1 Memoization vs Iteração sobre subproblemas

A palavra *memoization* vem do termo *memoize* que deriva de um termo latim *memorandum* que quer dizer lembrando. Essa é uma técnica *top-down* que armazena em uma tabela a solução de cada subsubproblema, evitando assim o retrabalho e aumentando a eficiência. Essa técnica utiliza um algoritmo recursivo.

A iteração sobre subproblemas é uma técnica *bottom-up* que utiliza um algoritmo iterativo para calcular os subproblemas. Essa técnica começa a calcular do menor subproblema para o maior.

2.2.3 Paradigma lógico

2.2.3.1 Surgimento da programação lógica

Os literais são fórmulas atômicas positivas ou negativas, que em conjunto formam uma cláusula. Um tipo de cláusula é a chamada cláusula de Horn (ou cláusula definitiva), onde cada cláusula tem no máximo um literal positivo. Essas cláusulas são utilizadas pela resolução SLD (Selection-rule driven Linear resolution for Definite clauses), que restringe o uso de literais para tornar as implementações mais eficientes. Essa resolução recebe como entrada um programa lógico P e uma consulta N , e efetua a resolução entre a consulta N e alguma regra de P . O ganho de eficiência vem do fato de que, para cada regra é possível a seleção de apenas um literal para a resolução com N (RODRIGUES, 2010). A programação lógica surgiu da aplicação da SLD no processamento da linguagem natural da inteligência artificial, e esse processamento serviu como primeira aplicação e motivo imediato para o desenvolvimento do PROLOG (Programation en Logique) (RODRIGUES, 2010).

2.2.3.2 PROLOG

O PROLOG é uma linguagem de programação utilizada para resolver problemas que envolvam objetos e relações entre objetos. O termo objeto em PROLOG não se refere a uma estrutura de dados que pode herdar variáveis e métodos de uma classe, mas se refere à coisas que podemos representar usando termos (CLOCK SIN; MELLISH, 2003). Um programa PROLOG consiste em um conjunto de cláusulas, onde cada cláusula ou é um fato sobre a informação dada ou uma regra sobre como a solução pode relacionar ou ser inferida a partir dos fatos dados (CLOCK SIN; MELLISH, 2003). Segundo (CLOCK SIN; MELLISH, 2003), a programação em PROLOG consiste em:

- especificar algum fato sobre os objetos e seus relacionamentos;
- definir algumas regras sobre objetos e seus relacionamentos;
- fazer perguntas sobre os objetos e seus relacionamentos.

O PROLOG pode fazer muito mais do que apenas responder sim ou não à perguntas feitas, ele permite que um computador seja usado como um armazém de fatos e regras, e fornece maneiras de se fazer inferências a partir de um fato ou outro e encontrar os valores das variáveis que levam a uma dedução lógica.

2.2.3.2.1 Fundamentos do PROLOG

- **Fato:** Um fato é a definição das relações entre os objetos. Para implementar essas definições algumas regras devem ser seguidas i) o nome de todas as relações e objetos devem começar com letra minúscula; ii) a relação é escrita primeiro seguida do(s) objeto(s) entre parênteses (caso tenha mais de um separar com vírgula); iii) um fato deve terminar com ponto-final. A relação entre os objetos é chamada de predicado, e os objetos da relação são chamados de argumento.
- **Questão:** a questão tem a escrita muito parecida com o fato, exceto que começa com um ponto de interrogação. Para cada pergunta feita ao PROLOG ele irá procurar na sua base de conhecimento fatos ou regras que comprovem aquele questionamento. Caso seja encontrada alguma prova, ele responde "sim", caso contrário, responde "não".
- **Variáveis:** é um nome com letra maiúscula, que quando utilizado numa pergunta faz com que o PROLOG procure na base de conhecimento um ou mais objetos que sejam equivalentes àquela variável. Como por exemplo, na base de conhecimento existem dois fatos:

```
gosta(maria, pedro).  
gosta(maria, joao).  
gosta(laura, joao).
```

Quando for perguntado ao prolog:

```
?- gosta(maria, X).
```

O PROLOG responderá:

```
X = pedro;  
X = joao.
```

- **Conjunção:** é o operador vírgula (,) tendo função de "e". Quando utilizado na pergunta, por exemplo, o PROLOG procura na base de conhecimento fatos que comprovem as duas perguntas. Usando a base de conhecimento do item anterior, se perguntar ao PROLOG:

```
?- gosta(maria, X) , gosta(laura, X).
```

ele responderá:

```
X = joao.
```

Para responder a esse questionamento o PROLOG procura primeiramente qual o objeto que pode ser a variável X, achando inicialmente o argumento pedro, logo após ele procura se existe algum fato que Laura goste de Pedro, como não existe esse fato na base de conhecimento, ele volta para buscar outro objeto que possa ser a variável X, e encontra o argumento joao. Ao buscar por algum fato que comprove que Laura gosta de João, ele encontra e retorna "sim" para a pergunta, ou seja, Maria gosta de João e Laura também. Caso ouvesse mais pessoas em comum que Maria e Laura gostassem era só digitar ";" seguido de Enter e o PROLOG buscaria por outros objetos para X.

- **Regras:** uma regra é uma declaração geral sobre objetos e seus relacionamentos. Em PROLOG, uma regra é comumente utilizada para dizer que um fato depende de um conjunto de outros fatos. Em PROLOG, uma regra contém uma cabeça e um corpo, ligados por pelo símbolo ":-", que é constituído por dois pontos e um hífen, e é pronunciado "se".

2.3 Considerações finais

3 Suporte Tecnológico

Para dar suporte ao desenvolvimento do projeto foi feito um levantamento de ferramentas candidatas, sendo este refinado e descrito a seguir.

3.1 Robótica Educacional

Um dos maiores precursores da robótica no âmbito educacional foi Seymour Papert ao criar a linguagem LOGO com o intuito de incentivar a aprendizagem da matemática, baseado nas idéias do suíço Jean Piaget que dizia, *"as funções essenciais da inteligência consistem em compreender e inventar, em outras palavras, construir estruturas estruturando o real"*, ou seja, é essencial para a formação da inteligência a ação sobre objetos, e por meio dessa descobrir propriedades por meio de abstração. Desde então, a robótica educacional vem se tornando uma plataforma atraente para criar envolvimento nos estudantes, incentivando o estudo da ciência e da tecnologia. As plataformas iterativas/lúdicas vem ganhando espaço ao longo dos anos como ferramentas auxiliares às metodologias de ensino que utilizam a robótica educacional. Dentre essas plataformas podemos citar: Robomind¹, Scratch², Arduino³, Raspberry Pi⁴ e o Kit Lego Mindstorms⁵. Essas plataformas serão brevemente descritas a seguir.

3.1.1 Robomind

O Robomind é uma IDE para a programação dos movimentos de um robô em um mundo bidimensional, através de uma linguagem de programação bem simples e intuitiva, ideal para iniciantes. A interface da IDE é composta por quatro partes: à cima contém o menu e os atalhos para as funções mais utilizadas, como inserir qualquer movimento; à esquerda encontra-se a área de edição do código; à direita contém o mundo bidimensional, onde é possível visualizar o robô realizando os movimentos, previamente determinados pelo código, em um mapa; e na parte inferior encontra-se o controle de execução do programa juntamente à área de mensagens utilizadas pela IDE para informar erros sintáticos ou durante a execução do código, e a situação na qual o robô se encontra. O Robomind é gratuito para teste durante 30 dias. Após esse período é possível compra-lo por USD 14.00 por ano. Ele pode ser instalado em vários idiomas, dentre eles o Português⁶, além

¹ www.robomindacademy.com

² www.scratch.mit.edu

³ www.arduino.cc

⁴ www.raspberrypi.org

⁵ mindstorms.lego.com

⁶ www.robomind.net/pt/

de ser multiplataforma, Linux, Mac e Windows. Com o Robomind só é possível simular o comportamento do que é implementado, não trabalhando a montagem de um robô, programação embarcada e utilização de sensores.

3.1.2 Scratch

O Scratch é um ambiente e uma linguagem de programação multimídia, produzido pelo *MediaLab*⁷ do MIT, para ser utilizado por crianças a partir dos oito anos de idade para criação de histórias iterativas, animações, jogos, músicas, dentre outros, ditos projetos Scratch. Um projeto Scratch contém personagens programáveis, que podem tratar eventos vindos do teclado ou do mouse, para mudança de posição e direção, dentre outras funcionalidades. Assim como o Robomind, ele utiliza uma linguagem de programação em blocos, com encaixe seletivo ideal para programadores iniciantes. O Scratch é gratuito, *online* e também tem a opção de utilizá-lo em português. Similar ao Robomind, no Scratch só é possível simular de forma limitada a implementação, não sendo possível testar em um robô físico nem utilizar sensores.

3.1.3 Arduino

O Arduino é uma plataforma de prototipagem eletrônica *open-source*, criada em 2005 pelo italiano Massimo Banzi, para auxiliar no ensino de eletrônica visando o baixo custo para os alunos. A plataforma é composta por um *hardware* e um *software* bastante flexíveis. O *hardware* é constituído de uma placa com um microprocessador; porta USB, usada para comunicação serial com o computador; pinos digitais, utilizados para detecção ou transmissão de controles digitais; pinos analógicos, usados para leitura de sinais de sensores; e pinos de alimentação, usados para alimentação de circuitos externos. O *software* é uma IDE, onde será programado o código, conhecido como *sketch*, e por meio desta será passado à placa através de uma comunicação serial. Nesta IDE é utilizada a linguagem de programação Arduino, mas quando é passado para a placa, esta linguagem é traduzida para a linguagem estruturada C. Sua IDE é multiplataforma, podendo ser instalada em Linux, Mac e Windows. O Arduino tem a vantagem de utilizar programação embarcada no robô e possuir entradas para diversos sensores, porém a montagem do robô fica a critério do desenvolvedor não tendo padronização alguma.

3.1.4 Raspberry Pi

O Raspberry Pi é um computador versátil e flexível, criado pela *Raspiberry Pi Foundation*⁸ em conjunto com a Universidade de Cambrigde, no ano de 2012, com o objetivo de estimular o ensino da ciência da computação em jovens do ensino básico. O

⁷ www.media.mit.edu

⁸ www.raspberrypi.org/about/

Raspberry Pi possui dois modelos, ambos equipados com processador multimídia *Broadcom BCM2835 system-on-chip* (SoC) de 700 Mhz com placa gráfica integrada VideoCore IV; entrada para cartão de memória, que se faz necessário pois o Raspberry Pi não possui armazenamento interno; interface HDMI; entrada USB e RCA; processador ARM 7 com capacidade de processamento de 32 bits, não sendo possível a instalação do Windows, mas aceitando qualquer distribuição Linux, como o Raspbian que é baseado no sistema operacional Debian. O que difere nas duas versões é a memória RAM. Um modelo é oferecido com 256 MB e o outro com 512 MB. Assim como o Arduino, a desvantagem do Raspberry Pi é a não padronização da montagem dos robôs. Entretanto, é possível a utilização de sensores e programação embarcada.

3.1.5 Kit Educacional Lego Mindstorms

A LEGO lançou em janeiro de 2006, na feira *Consumer Electronics Show*⁹ em Las Vegas, a linha Mindstorms NXT, que é uma versão mais avançada que a já consagrada RCX, possuindo um processador Atmel ARM 32 bits com *clock* de 48MHz, HD de 256KB de memória *flash*, memória RAM de 64KB, software próprio, sensores de luz, toque e som. Esse suporte confere ao robô noções de distância, sendo inclusive capaz de reagir a movimentos, ruídos e cores. Essa nova versão também executa movimentos com maior grau de precisão que seu antecessor. O kit Lego Mindstorms é bem flexível quanto a montagem do robô, porém a Lego disponibiliza um manual de montagem de um robô padrão, uma vantagem ao se comparar com os concorrentes mencionados anteriormente. A escolha do kit Lego Mindstorms para o dado projeto se deu, além da vantagem já conhecida da padronização do robô, por ele já conter os sensores necessários no kit e por esse material ser utilizado na matéria Princípios de Robótica Educacional, ministrada na FGA pelo prof. Dr. Maurício Serrano, co-orientador deste projeto.

FALTA COLOCAR O QUE SERÁ UTILIZADO PARA RODAR O CÓDIGO EM PROLOG NO ROBÔ.

3.2 Engenharia de Software

3.2.1 Ubuntu

O Linux foi criado por Linus Torvalds em 1991 ([TORVALDS; BY-DIAMOND, 2001](#)) e desde então amplamente aderido pela comunidade. Com base nesse kernel vários outros sistemas operacionais foram criados, dentre eles o Ubuntu. Baseado no Debian e patrocinado pela Canonical Ltda., o Ubuntu é licenciado por General License Public (GPL), podendo ser então modificado e distribuído de acordo com os termos da licença.

⁹ www.cesweb.org

Robustez e confiabilidade, por meio de código aberto, compatibilidade, segurança, rapidez e robustez são algumas características que o fazem tão popular.

3.2.2 SWIProlog

O SWIProlog é uma implementação, de código aberto, para a linguagem de programação Prolog, executando em modo texto, através de comandos no terminal do sistema. Sob a licença *Lesser GNU Public License*¹⁰, pode ser utilizado nas plataformas Windows, Linux e MacOS. Possui diversas ferramentas de edição gráfica, tais como: J-Prolog Editor e SWI-Prolog-Editor. Permite ainda a utilização da linguagem Prolog por outras linguagens, tais como: C/C++ e Java.

3.2.3 Bizagi Process Modeler

O BizAgi Process Modeler¹¹ é um aplicativo gratuito utilizado para criar e documentar modelos de processos em BPMN¹² (*Business Process Model and Notation*). Esse aplicativo faz parte de uma suíte de software, chamada Bizagi, composta por dois produtos: o Bizagi Process Modeler e o BizAgi BPM Suite.

3.2.4 Git

O Git¹³ é um dos mais consagrados e utilizados sistemas de controle de versão distribuída. É *open-source* e gratuito, distribuído sob a licença GNU GPLv2¹⁴, e projetado para lidar com qualquer tamanho de projeto mantendo a rapidez e eficiência.

3.2.5 Github

O GitHub é um repositório web planejado para utilizadores do sistema de controle de versão Git. Possui opções de utilização de repositórios privados e públicos, sendo o pago e gratuito, respectivamente. No GitHub, é possível, dentre tantas outras funcionalidades, visualizar o código no navegador, criar *issues*, revisar e aceitar mudanças, baixar o repositório como *.zip*, criar *branches* ou até mesmo fazer o *fork* de outro repositório.

3.2.6 Waffle

O Waffle¹⁵ é uma solução de gerenciamento de projeto *online*, que se integra ao repositório do GitHub e cria um *board*, muito semelhante a um *kanban*, com as *issues*

¹⁰ <http://www.gnu.org/licenses/lgpl.html>

¹¹ <http://www.bizagi.com/en/bpm-suite/bpm-products/modeler>

¹² www.bpmn.org

¹³ <http://www.git-scm.com/>

¹⁴ <http://www.gnu.org/licenses/gpl-2.0.html>

¹⁵ www.waffle.io

que estão cadastradas, e as agrupa de acordo com seus status, que pode ser Backlog, Ready, In Progress e Done. Cada *issue*, se estiver associada a um Milestone no GitHub, será também associada a um Milestone no Waffle. Qualquer alteração feita no GitHub é automaticamente compatibilizada no Waffle.

3.2.7 LaTeX

O Latex¹⁶ é um sistema de preparação de documentos para composição tipográfica de alta qualidade, desenvolvido inicialmente por Leslie Lamport, em 1985, baseado na linguagem TeX criada por Donald E. Knuth, no final da década de 70, na Universidade de Stanford. Atualmente, o Latex é mantido e desenvolvido pelo *The Latex3 Project*¹⁷, sendo disponibilizado gratuitamente. O Latex é mais amplamente utilizado no meio técnico-científico para escrita de documentos de médio a grande porte, devido a sua facilidade de produzir fórmulas e símbolos matemáticos.

3.2.8 TexMaker

O TeXMaker¹⁸ é um editor de texto Latex gratuito, multiplataforma para Linux e MacOSX, com suporte a unicode, verificação ortográfica, auto-completar e contém um visualizador de pdf embutido. Com uma interface simples, o TexMaker contém botões de atalho para as funções mais utilizadas como: estruturas (part/chapter/section), referências (ref/cite), tamanho e estilo de letra, negrito, itálico, alinhar (direta/esquerda, centro) e inserir diversos símbolos matemáticos.

3.3 Considerações finais

¹⁶ <http://www.latex-project.org/>

¹⁷ <http://latex-project.org/latex3.html>

¹⁸ <http://www.xmlmath.net/texmaker/>

4 Metodologia

4.1 Classificação da pesquisa

As pesquisas podem ser classificadas quanto ao seus objetivos gerais ou quanto aos procedimentos técnicos utilizados. A classificação das pesquisas quanto aos seus objetivos gerais é muito importante para estabelecer uma visão teórica, ou seja, possibilitar uma aproximação conceitual. Porém, é necessário confrontar essa visão teórica com dados realistas, fornecendo uma visão empírica. Desta forma, faz-se necessário traçar um modelo conceitual e operativo, chamado delineamento, e que tem como parte mais importante a definição do procedimento que será utilizado na coleta de dados. Logo, as pesquisas podem ser classificadas de acordo com seu delineamento, ou seja, quanto aos procedimentos técnicos utilizados (GIL, 2002).

4.1.1 Quanto aos Objetivos Gerais

As pesquisas classificadas pelos objetivos gerais podem ser divididas em 3 (três) grandes grupos: pesquisas exploratórias, pesquisas descritivas e pesquisas explicativas (GIL, 2002). Cada um desses grupos será detalhado nas subseções a seguir apresentadas.

4.1.1.1 Pesquisa Exploratória

A pesquisa exploratória tem um caráter investigativo sobre um assunto pouco conhecido. Tem por objetivo facilitar a delimitação do tema de pesquisa através de informações proporcionadas pela exploração (PRODANOV; FREITAS, 2013). Por isso, seu planejamento é bastante flexível, de modo que considere os mais diversos aspectos acerca do tema estudado. Em grande maioria, as pesquisas exploratórias assumem a forma de pesquisa bibliográfica ou estudo de caso, envolvendo levantamento bibliográfico, entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado e/ou análise de exemplos que estimulem a compreensão (GIL, 2002).

4.1.1.2 Pesquisa Descritiva

Na pesquisa descritiva, o pesquisador apenas observa, registra, analisa e ordena os dados dos fatos observados sem interferir de forma alguma no processo. Envolve o uso padronizado de coleta de dados podendo ser questionário e/ou observação sistemática. Em geral, assume a forma de levantamento (GIL, 2002). A pesquisa descritiva tem por objetivo descrever as características de determinada população, fenômeno ou estabelecimento de relações entre variáveis procurando classificar, explicar e interpretar os fatos

que ocorrem, diferindo da pesquisa experimental que pretende demonstrar o modo ou as causas de um dado fato ocorrido (PRODANOV; FREITAS, 2013).

4.1.1.3 Pesquisa Explicativa

A pesquisa explicativa, assim como o nome sugere, busca explicar os motivos pelos quais ocorrem os fatos observados, por meio do registro, análise, classificação e interpretação dos fenômenos observados. Em geral, quando realizada nas ciências naturais, requer o uso do método experimental que possibilita a manipulação e controle de variáveis, e quando realizada nas ciências sociais, requer o uso do método observacional. Dessa forma, a pesquisa explicativa assume a forma de pesquisa experimental ou pesquisa *ex-post facto*. (PRODANOV; FREITAS, 2013)

4.1.2 Quanto aos Procedimentos

As pesquisas classificadas quanto aos procedimentos técnicos utilizados podem ser divididas em 2 (dois) grupos: pesquisas que se valem das fontes de "papel", sendo elas classificadas em bibliográfica ou documental; e as pesquisas cujos dados são fornecidos por pessoas, sendo classificadas em experimental, *ex-post facto*, levantamento ou estudo de caso (GIL, 2002). Cada uma dessas classificações será detalhada a seguir.

4.1.2.1 Pesquisa Bibliográfica

Uma pesquisa bibliográfica é desenvolvida exclusivamente a partir de fontes bibliográficas, principalmente de livros e artigos científicos. As fontes bibliográficas podem ser classificadas de acordo com a Figura 3.

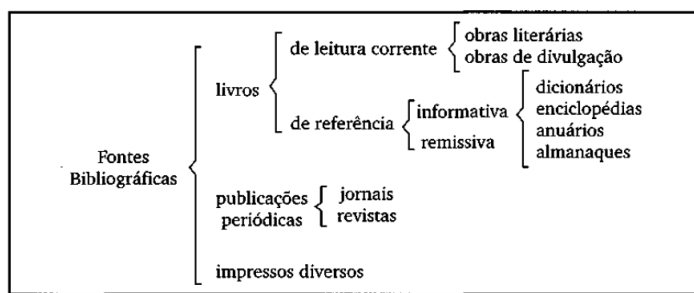


Figura 3 – Classificação das pesquisas bibliográficas (GIL, 2002)

Os livros são fontes bibliográficas de excelência, classificados como de leitura corrente ou de referência. Os livros de literatura corrente são aqueles que objetivam conferir conhecimentos científicos ou técnicos ao leitor, já os de referência tem por objetivo proporcionar ao leitor a rápida obtenção das informações requeridas (informativa) ou a referência às obras que as contenham (remissiva) (GIL, 2002). A pesquisa bibliográfica é muito vantajosa ao pesquisador, já que este pode ter acesso a uma gama de informações muito maior

do que aquelas que ele conseguiria pesquisando diretamente. Essa pesquisa também é de suma importância nas pesquisas históricas, pois a maioria dos fatos só é possível conhecer por esse tipo de pesquisa. A desvantagem dessa pesquisa é a possível propagação de erros, pois se a fonte bibliográfica utilizada estiver com algum dado equivocado, este será refletido no trabalho do pesquisador.

4.1.2.2 Pesquisa Documental

A pesquisa documental é difícil de distinguir da pesquisa bibliográfica, pois ambas utilizam material impresso para um determinado público como fonte de pesquisa. A principal diferença que se pode destacar é quanto ao tratamento analítico das fontes. A pesquisa documental utiliza fontes de pesquisa que não passaram por um tratamento analítico, ou seja, são matéria-prima como cartas pessoais, fotos, diários, memorandos e outros, a partir da qual o pesquisador vai desenvolver a sua pesquisa. A pesquisa bibliográfica se utiliza de fontes de pesquisa já analisadas como livros e periódicos (GIL, 2002).

4.1.2.3 Pesquisa Experimental

A pesquisa experimental consiste em determinar o objeto de estudo, as possíveis variáveis capazes de influenciá-lo, as formas de controle e observação dos efeitos, fazendo com que o pesquisador seja o agente ativo da pesquisa. Uma pesquisa experimental pode ser desenvolvida em qualquer lugar, desde que ela possa ser manipulada e controlada pelo pesquisador e que haja uma distribuição aleatória dos elementos que irão participar do experimento. As pesquisas experimentais são as mais utilizadas para se provar hipóteses de causa-efeito, porém é um ambiente com variáveis controladas, logo, pode haver variáveis de difícil controle ou até mesmo impossível, tornando o experimento inviável. (GIL, 2002)

4.1.2.4 Pesquisa *Ex-post Facto*

A tradução literal de *ex-post facto* é "de um fato passado", ou seja, a pesquisa *ex-post facto* é realizada após a ocorrência de variáveis no objeto de estudo. A diferença entre a pesquisa *ex-post facto* e a pesquisa experimental é que esta verifica a causa-efeito com o controle de variáveis, enquanto aquela verifica a causa-efeito sem o controle de variáveis, já que o fato já ocorreu. Esse tipo de pesquisa é muitas vezes chamada de correlacional, pois nela é possível se identificar a existência de relação entre variáveis. (GIL, 2002)

4.1.2.5 Levantamento

O levantamento é utilizado quando se interroga diretamente as pessoas que se quer conhecer o comportamento, procura ser representativo de universo definido, identificando as características dos componentes desse universo e oferecendo resultados pela precisão

estática, através da caracterização dos segmentos. Um questionário pode ser utilizado para coletar os dados, e depois é feita a análise quantitativa dessa coleta. Quando um levantamento é feito sobre um grupo muito grande de pessoas, este é chamado de "censo". (PRODANOV; FREITAS, 2013)

4.1.2.6 Estudo de Caso

O estudo de caso busca o aprofundamento nas questões propostas, estudando um único grupo ou comunidade, utilizando mais a observação direta do que a interrogação, captando as explicações e interpretações do que ocorre no grupo. No estudo de campo, o pesquisador está inserido no grupo, para poder entender melhor as regras, os costumes e as convenções.

4.2 Planejamento da pesquisa

O modelo híbrido de pesquisa exploratório/experimental será utilizado nesse TCC para avaliar os vários tipos de algoritmos que podem ser utilizados como solução para a questão de pesquisa. Após a escolha do algoritmo ter sido feita, este será adaptado ao contexto do LEGO Mindstorms, utilizando o modelo híbrido de pesquisa exploratório/-descritivo. Pretende-se ainda, nas atividades específicas de desenvolvimento, fazer uso de uma metodologia orientada aos princípios ágeis, uma possível adaptação do Scrum, com sprints de uma semana (em média), as quais ocorrerão em ambiente controlado e terão seus resultados quantificados. Dado que a pesquisa será exploratória/experimental e exploratória/descritiva, muito provavelmente será utilizado o modelo de pesquisa-ação ao longo do projeto. Esse modelo permitirá, dentre outras contribuições, a retroalimentação da solução com base nos experimentos realizados ao longo da condução do trabalho.

4.3 Modelagem da metodologia

A modelagem da metodologia utilizada neste trabalho demonstra, de forma clara, o fluxo das atividades tanto do trabalho de conclusão de curso 1 quanto do trabalho de conclusão de curso 2, visto que um é a continuação do outro. Inicialmente foi feito um levantamento bibliográfico inicial sobre robótica, algoritmos de otimização, como o algoritmo guloso, e programação dinâmica, a fim de dar suporte a definição do escopo do projeto. A seguir, foi realizado um planejamento da abordagem para definir se a mesma seria ou não viável. Após a aprovação do escopo, foi definido um referencial teórico, um suporte tecnológico e uma metodologia de pesquisa. Concluídos esses passos, foi implementada a prova de conceito e então elaborada a parte escrita da proposta. A apresentação para a banca avaliadora findará a fase do TCC 01, dando início ao TCC 02. Este começa com a coleta e realização das sugestões de melhoria feitas pela banca

avaliadora, a seguir são definidos cenários de uso que serão utilizados para avaliar os resultados da implementação da máquina de raciocínio. Em sprints de uma semana, durante três meses, a máquina de raciocínio será desenvolvida. A cada duas sprints, uma release, a máquina de raciocínio será testada no robô e seus resultados serão avaliados de acordo com os cenários de uso anteriormente definidos. Caso o resultado não seja satisfatório, uma nova release será necessária. Quando a avaliação dos resultados for satisfatória será elaborada a parte escrita do projeto. A definição temporal de uma release ter duas sprints e cada sprint ter duas semanas é uma média, podendo haver sprints ou releases maiores ou menores, de acordo com a necessidade. Uma release é dada por encerrada quando o código é testado no robô e seus resultados são avaliados. Com intuito de ilustrar como será realizada a condução deste projeto, segue a Figura 4 que apresenta a modelagem do processo metodológico com base na metodologia descrita nesta seção.

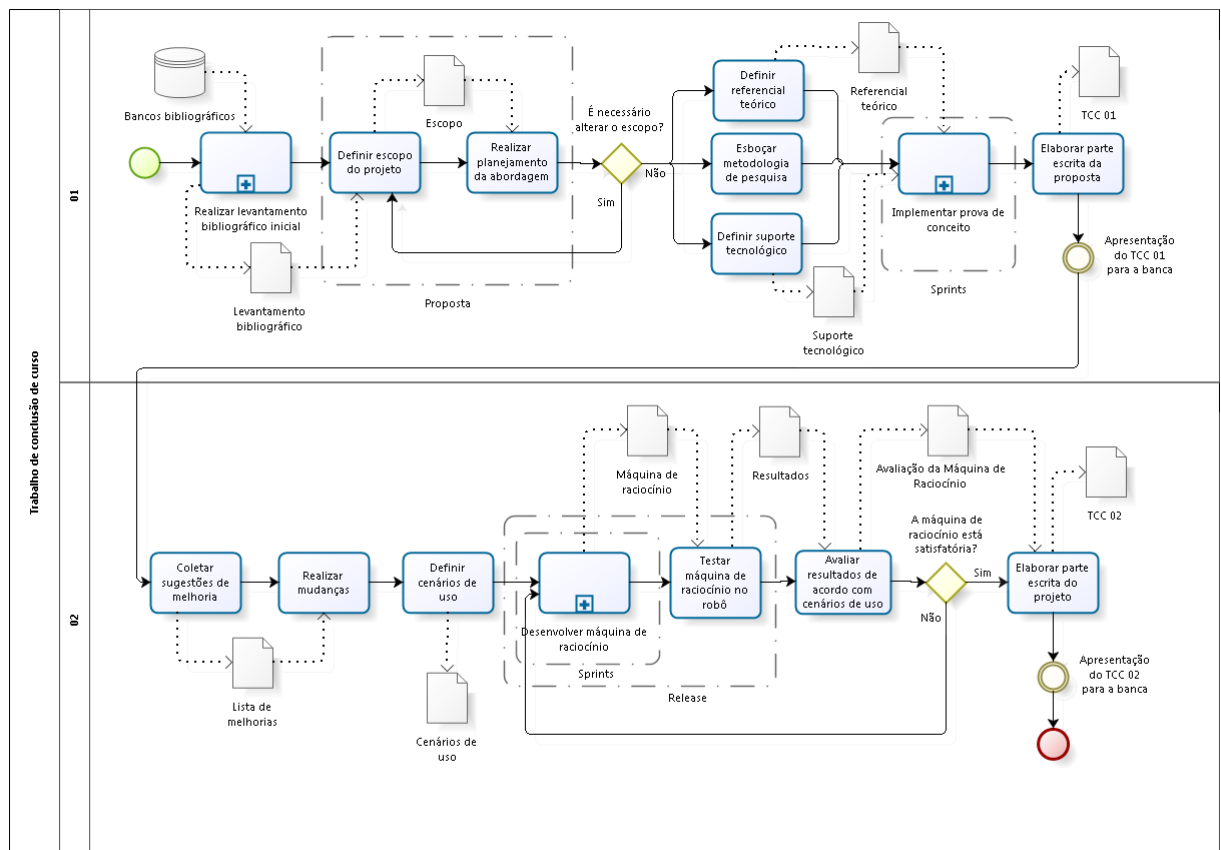


Figura 4 – Modelagem do processo utilizado no TCC

4.4 Cronograma da pesquisa

O cronograma, detalhado nas Tabelas 1 e 2, visa conferir uma noção temporal acerca das atividades definidas para o processo. Trata-se de uma visão preliminar, logo, pode haver modificações ao longo do projeto de acordo com as necessidades.

	Jan	Fev	Mar	Abr	Mai	Jun
Realizar levantamento bibliográfico inicial	X	X				
Definir escopo do projeto		X				
Realizar planejamento da abordagem			X			
Definir referencial teórico				X	X	
Definir suporte tecnológico				X	X	
Esboçar metodologia de pesquisa				X	X	
Implementar prova de conceito						X
Elaborar parte escrita da proposta						X

Tabela 1 – Cronograma do TCC 01

	Jul	Ago	Set	Out	Nov	Dez
Coletar sugestões de melhoria	X					
Realizar mudanças	X					
Definir cenários de uso	X					
Desenvolver máquina de raciocínio		X	X	X		
Testar máquina de raciocínio no robô			X	X		
Avaliar resultados de acordo com cenários de uso			X	X		
Elaborar parte escrita do projeto					X	

Tabela 2 – Cronograma do TCC 02

4.5 Considerações finais

5 Prova de conceito

5.1 Descrição da prova de conceito

5.2 Considerações finais

6 Proposta

6.1 Descrição da proposta

6.2 Considerações finais

7 Resultados Obtidos

8 Conclusão

Referências

- ANTONIO, B. et al. Fundamentos de robótica. *Ed. Mc Graw Hill*, 1997. Citado 2 vezes nas páginas 35 e 36.
- BENITTI, F. B. V. et al. Experimentação com robótica educativa no ensino médio: ambiente, atividades e resultados. In: *Anais do XXIX Congresso da Sociedade Brasileira de Computação, Bento Gonçalves/RS*. [S.l.: s.n.], 2009. p. 1811–1820. Citado na página 39.
- CASTALONGA, J. P.; LEMOS, J. M. S. et al. Cobertura e empacotamento por circuitos através de um elemento em matróides. Universidade Federal de Pernambuco, 2007. Citado na página 47.
- CLOCKSIN, W.; MELLISH, C. S. *Programming in PROLOG*. [S.l.]: Springer Science & Business Media, 2003. Citado na página 49.
- CORMEN, T. H. *Introduction to algorithms*. [S.l.]: MIT press, 2009. Citado na página 47.
- CURCIO, C. P. d. C. Instituto de tecnologia para o desenvolvimento (lactec) instituto de engenharia do paran  (iep) programa de p s-gradua  o em desenvolvimento de tecnologia (prodetec). 2008. Citado na p gina 36.
- GIL, A. C. *Como elaborar projetos de pesquisa*. [S.l.]: S o Paulo (SP): Atlas, 2002. Citado 4 vezes nas p ginas 17, 57, 58 e 59.
- GRASSI, V. Arquitetura h brida para rob s m veis baseada em fun  es de navega  o com intera  o humana. *Escola Polit cnica, USP.*, 2006. Citado na p gina 44.
- INDUSTRIAIS, E. *SISTEMA DE NAVEGA  O AUT NOMA PARA ROB S M VEIS BASEADO EM ARQUITETURA H BRIDA: TEORIA E APLICA  O*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DE ITAJUB , 2009. Citado na p gina 44.
- KAMEN, D.; KRISTIANSEN, K. K. First lego league. 2010. Citado na p gina 30.
- MURPHY, R. *Introduction to AI robotics*. [S.l.]: MIT press, 2000. Citado na p gina 29.
- PIO, J. L. de S.; CASTRO, T. H. C. de; J NIOR, A. N. de C. A rob tica m vel como instrumento de apoio   aprendizagem de computa  o. In: *Anais do S mp sio Brasileiro de Inform tica na Educa  o*. [S.l.: s.n.], 2006. v. 1, n. 1, p. 497–506. Citado na p gina 36.
- PRODANOV, C. C.; FREITAS, E. C. de. *Metodologia do Trabalho Cient fico: M todos e T cnicas da Pesquisa e do Trabalho Acad mico-2  Edi  o*. [S.l.]: Editora Feevale, 2013. Citado 3 vezes nas p ginas 57, 58 e 60.
- ROCHA, A. Algoritmos gulosos: defini  es e aplica   es. Campinas, SP, 2004. Citado 2 vezes nas p ginas 45 e 46.

- RODRIGUES, T. G. *Sobre os Fundamentos da Programação Lógica Paraconsistente*. Tese (Doutorado) — Universidade Estadual de Campinas, 2010. Citado na página 48.
- RUSSEL, S.; NORVIG, P. Inteligência artificial. *Editora Campus*, 2004. Citado na página 29.
- SANTOS, C. C. d. Robótica na construção: uma aplicação prática. Universidade de Coimbra, 2002. Citado 2 vezes nas páginas 35 e 36.
- SILVA, A. F. D. *RoboEduc: Uma metodologia de aprendizado com Robótica Educacional*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE, 2009. Citado na página 29.
- SILVA, A. F. de et al. Utilização da teoria de vygotsky em robótica educativa, em ‘. In: IX CONGRESO IBEROAMERICANO DE INFORMATICA EDUCATIVA RIBIE. [S.l.], 2008. Citado 2 vezes nas páginas 35 e 36.
- TORVALDS, L.; BY-DIAMOND, D. R. *Just for fun: The story of an accidental revolutionary*. [S.l.]: Harper Audio, 2001. Citado na página 53.
- TUCKER, A.; NOONAN, R. *Linguagens de Programação-: Princípios e Paradigmas*. [S.l.]: Grupo A Educação, 2009. Citado na página 31.
- WOLF, D. F. et al. Robótica móvel inteligente: Da simulação às aplicações no mundo real. In: *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*. [S.l.: s.n.], 2009. Citado 5 vezes nas páginas 17, 41, 42, 43 e 44.
- ZILLI, S. d. R. et al. A robótica educacional no ensino fundamental: perspectivas e prática. Florianópolis, SC, 2004. Citado 2 vezes nas páginas 39 e 40.

Apêndices

APÊNDICE A – Primeiro Apêndice

Texto do primeiro apêndice.

APÊNDICE B – Segundo Apêndice

Texto do segundo apêndice.

Anexos

ANEXO A – Primeiro Anexo

Texto do primeiro anexo.

ANEXO B – Segundo Anexo

Texto do segundo anexo.