

**UNIVERSIDADE FEDERAL FLUMINENSE
ANDRÉ CARLOS TEIXEIRA VASCONCELOS**

**SISTEMA INTELIGENTE DE APOIO À DECISÃO: UM ESTUDO
SOBRE ALGORITMO GENÉTICO E LÓGICA FUZZY**

Niterói

2016

ANDRÉ CARLOS TEIXEIRA VASCONCELOS

**SISTEMA INTELIGENTE DE APOIO À DECISÃO: UM ESTUDO
SOBRE ALGORITMO GENÉTICO E LÓGICA FUZZY**

Trabalho de Conclusão de Curso
submetido ao Curso de Tecnologia em
Sistemas de Computação da
Universidade Federal Fluminense como
requisito parcial para obtenção do título
de Tecnólogo em Sistemas de
Computação.

**Orientadora:
Mariana Tasca**

**NITERÓI
2016**

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

V331 Vasconcelos, André Carlos Teixeira
Sistema inteligente de apoio à decisão : um estudo sobre
algoritmo genético e lógica fuzzy / André Carlos Teixeira
Vasconcelos. – Niterói, RJ : [s.n.], 2016.
107 f.

Projeto Final (Tecnólogo em Sistemas de Computação) –
Universidade Federal Fluminense, 2016.
Orientador: Mariana Tasca.

1. Sistema inteligente (Ciência da Computação). 2. Algoritmo
genético. 3. Lógica fuzzy. 4. Processo decisório. I. Título.

CDD 006.33

ANDRÉ CARLOS TEIXEIRA VASCONCELOS

**SISTEMA INTELIGENTE DE APOIO À DECISÃO: UM ESTUDO
SOBRE ALGORITMO GENÉTICO E LÓGICA FUZZY**

Trabalho de Conclusão de Curso
submetido ao Curso de Tecnologia em
Sistemas de Computação da
Universidade Federal Fluminense como
requisito parcial para obtenção do título
de Tecnólogo em Sistemas de
Computação.

Niterói, ____ de _____ de 2016.

Banca Examinadora:

Prof^a. Mariana Tasca Fontenelle Lôbo, DSc. – Orientadora
UFF – Universidade Federal Fluminense

Prof^a. Juliana M N S Zamith, DSc. – Avaliadora
UFF – Universidade Federal Fluminense

Dedico este trabalho: Ao Grande Arquiteto do Universo. À memória de meu querido pai, a minha mãe e ao meu irmão. Ao amor de minha vida. Ao meu filho do coração. Aos meus amigos. E a todos que sempre nos ajudam a seguir em frente.

AGRADECIMENTOS

Ao grande Arquiteto de Universo, no qual me concebeu esta oportunidade única de estudar em uma instituição de prestigiado nome, que sempre me guiou, iluminou e protegeu por todo o caminho nesta longa jornada.

A minha orientadora Prof.^a Mariana Tasca por acreditar em mim, pelo estímulo e atenção na explanação de idéias, pelas orientações e nas palavras de motivação que conduziu minha orientação nesses últimos meses.

Aos meus amigos André Silva Aguiar, Rodolfo Alves de Oliveira e Marcelo Pitanga Alves por todo o incentivo, a confiança, o apoio e pela ajuda com o fornecimento de material para a realização deste trabalho.

A todos os meus familiares pelo incentivo, apoio e colaboração.

“A coisa mais indispensável a um homem é reconhecer o uso que deve fazer do seu próprio conhecimento”.

Platão, filósofo grego.

RESUMO

O principal objetivo deste trabalho é o de realizar um estudo sobre a aplicação de técnicas da inteligência artificial. Os sistemas de informação são ferramentas de grande importância na abordagem para a resolução de problemas, são classificados de acordo com o seu foco de atuação e por suas características. O conceito do algoritmo genético vem da inspiração da teoria de Darwin, que consiste de um processo evolutivo, no qual é iniciado com uma população de soluções, e processos de seleção natural são aplicados consecutivamente formando uma nova geração melhor que a anterior. A lógica fuzzy é baseada no raciocínio humano, refletindo a forma como pensamos, tentando modelar uma resposta mais adequada à realidade. As técnicas de algoritmo genético e da lógica fuzzy podem tornar um sistema mais eficiente e especializado em resolver problemas que possuam elevado grau de dificuldade, tornando-o mais eficiente para lidar com as restrições impostas no melhor tempo de execução e realizar buscas precisas indicando as melhores soluções.

Palavras-chaves: algoritmo genético, lógica fuzzy e apoio à decisão.

ABSTRACT

The main objective of this work is to conduct a study on the application of techniques of artificial intelligence. Information systems are very important tools in the approach to problem solving, they are classified according to its focus and its features. The genetic algorithm concept comes from the inspiration of Darwin's theory, which consists of an evolutionary process, which starts with a population of solutions, and natural selection processes are applied sequentially forming a new generation better than the last. Logic fuzzy is based on human reasoning, reflecting the way they think, trying to model a more appropriate response to reality. The genetic algorithm and fuzzy logic techniques can become a more efficient and specialized system to solve problems that have high degree of difficulty, making it more efficient to deal with the restrictions imposed in the best run time and perform precise searches indicating the best solutions.

Key words: genetic algorithm, fuzzy logic and decision support.

LISTA DE ILUSTRAÇÕES

Figura 1: Transformação dos dados pelos Sistemas de Informação.....	22
Figura 2: Atividades básicas de um Sistema de Informação.....	23
Figura 3: Processo de resolução de problemas.....	24
Figura 4: Classificação dos SI.....	26
Figura 5: Evolução da solução de problemas.....	29
Figura 6: Estrutura de um SAD.....	33
Figura 7: Estrutura de um SBC.....	38
Figura 8: Estrutura de um SAD Inteligente.....	46
Figura 9: Cruzamento entre ervilhas lisas e rugosas.....	50
Figura 10: Cruzamento entre ervilhas híbridas lisas.....	51
Figura 11: Cruzamento entre ervilhas lisas e amarelas com rugosas e verdes.....	52
Figura 12: Cruzamento entre ervilhas híbridas lisas e amarelas.....	52
Figura 13: Célula, Cromossomo e DNA.....	54
Figura 14: Exemplo de conversão gênica.....	55
Figura 15: Processo de crossing-over.....	56
Figura 16: Estrutura de funcionamento de um AG.....	60
Figura 17: População de quatro indivíduos.....	61
Figura 18: Avaliação dos indivíduos.....	62
Figura 19: Método da Roleta.....	63
Figura 20: Seleção Local.....	64
Figura 21: Método do Torneio.....	64
Figura 22: Método da Amostragem Universal Estocástico.....	65
Figura 23: Método da Seleção Truncada.....	66
Figura 24: <i>Crossover</i> de um ponto.....	66
Figura 25: <i>Crossover</i> Multipontos.....	67
Figura 26: <i>Crossover</i> Uniforme.....	68
Figura 27: <i>Crossover</i> Baseado em Maioria.....	68
Figura 28: Mutação.....	69

Figura 29: Diagrama típico de um modelo de inferência Mamdani.	84
Figura 30: Processo de Fuzzificação do dado de entrada.....	87
Figura 31: Processo de Inferência da relação velocidade e consumo.	87
Figura 32: Processo de Defuzzificação da informação obtida.....	88
Figura 33: Plataforma tecnológica da empresa.	90
Figura 34: Retorno da aplicação com os pedidos selecionados.....	93

LISTA DE TABELAS

Tabela 1: Dados, Informação e Conhecimento.	23
Tabela 2: Sistemas de Apoio Operacional.	27
Tabela 3: Sistemas de Apoio Gerencial.	28
Tabela 4: Sistemas de Processamento Especializado.	28
Tabela 5: Principais diferenças entre SI Convencionais e SBC.	37
Tabela 6: Benefícios da Integração de Sistemas.	45
Tabela 7: Terminologia utilizada em AG.	58
Tabela 8: Esquemas.	72
Tabela 9: Lista de pedidos para a rota 05.	92

LISTA DE GRÁFICOS

Gráfico 1: Desempenho do AG.	73
Gráfico 2: Gráfico representando os três conjuntos <i>fuzzy</i> no universo U.	79
Gráfico 3: Conjunto <i>fuzzy</i> resultante da operação de complemento.	80
Gráfico 4: Conjunto <i>fuzzy</i> resultante da operação de união.	80
Gráfico 5: Conjunto <i>fuzzy</i> resultante da operação de interseção.	81

LISTA DE ABREVIATURAS E SIGLAS

AG – Algoritmo Genético

GC – Gestão de Conhecimento

IA – Inteligência Artificial

LF – Lógica Fuzzy

RC – Representação de Conhecimento

SAD – Sistema de Apoio à Decisão

SADE – Sistema de Apoio à Decisão Executiva

SADG – Sistema de Apoio à Decisão em Grupo

SBC – Sistema Baseado em Conhecimento

SE – Sistema Especialista

SI – Sistema de Informação

SIAD – Sistema Inteligente de Apoio à Decisão

SIG – Sistemas de Informação Gerencial

SUMÁRIO

1	INTRODUÇÃO	17
1.1	MOTIVAÇÕES	17
1.2	CONTEXTUALIZAÇÃO	18
1.3	METODOLOGIA.....	20
1.4	ORGANIZAÇÃO DO TRABALHO	20
2	SISTEMAS DE INFORMAÇÃO	22
2.1	CONCEITOS E DEFINIÇÕES.....	22
2.2	RESOLUÇÃO DE PROBLEMAS.....	24
2.3	CLASSIFICAÇÃO DOS SISTEMAS	25
2.3.1	Sistemas de Apoio Operacional.....	27
2.3.2	Sistemas de Apoio Gerencial.....	27
2.3.3	Sistemas de Processamento Especializado	28
2.4	EVOLUÇÃO	29
2.5	SISTEMA DE APOIO À DECISÃO	30
2.5.1	Características e Capacidades	31
2.5.2	Estrutura e Componentes.....	32
2.5.3	Categorias Especiais	33
2.5.4	Modelagem e Modelos	35
2.6	SISTEMAS BASEADOS EM CONHECIMENTO.....	36
2.6.1	Diferente dos Demais	37
2.6.2	Estrutura e Componentes.....	38
2.6.3	Núcleo do Sistema.....	39
2.6.4	Conhecimento e sua Representação.....	40
2.7	INTEGRAÇÃO ENTRE SISTEMAS	44
3	ALGORITMOS GENÉTICOS	48
3.1	GENÉTICA.....	48
3.1.1	A Seleção Natural de Darwin.....	48

3.1.2	Leis da hereditariedade	49
3.1.3	DNA e Cromossomos	53
3.1.4	Recombinação Genética.....	55
3.1.5	Mutação.....	56
3.2	HISTÓRICO	57
3.3	DEFINIÇÃO E CARACTERÍSTICAS	58
3.4	ESTRUTURA BÁSICA	60
3.4.1	População Inicial.....	61
3.4.2	Avaliação	62
3.4.3	Seleção.....	63
3.4.4	Recombinação.....	66
3.4.5	Mutação.....	68
3.4.6	Parâmetros de Controle.....	70
3.4.7	Critério de Parada.....	70
3.5	TEOREMA DOS ESQUEMAS.....	71
3.6	ELITISMO.....	72
4	LÓGICA FUZZY	75
4.1	CONCEITO	75
4.2	HISTÓRICO	76
4.3	CONJUNTOS FUZZY	77
4.3.1	Operações <i>Fuzzy</i>	79
4.4	REPRESENTAÇÃO FUZZY DE CONHECIMENTO	81
4.4.1	Variável Lingüística.....	82
4.4.2	Regras de Produção <i>Fuzzy</i>	83
4.5	INFERÊNCIA FUZZY COM MAMDANI.....	83
4.5.1	Módulo de Interface de Entrada.....	84
4.5.2	Máquina de Inferência	85
4.5.3	Base de Regras	85
4.5.4	Módulo de Interface de Saída.....	85
4.6	LÓGICA FUZZY E OS SISTEMAS DE DECISÃO.....	86
5	APLICAÇÃO.....	89
5.1	CENÁRIO	89
5.2	PLATAFORMA TECNOLÓGICA.....	90
5.3	SOLUÇÃO PROPOSTA.....	91

5.4	SOLUÇÃO EM EXECUÇÃO	92
5.5	BENEFÍCIOS ADQUIRIDOS	94
6	CONCLUSÃO	95
6.1	SUGESTÕES PARA TRABALHOS FUTUROS	95
	REFERÊNCIAS BIBLIOGRÁFICAS	97
	APÊNDICE	101

1 INTRODUÇÃO

Neste capítulo apresentam-se as motivações para a elaboração do estudo proposto, as metodologias e técnicas aplicadas, a linguagem utilizada, ou seja, uma introdução geral sobre os próximos capítulos deste trabalho, sua organização e seus objetivos.

1.1 MOTIVAÇÕES

Nesta seção apresentam-se as principais motivações que influenciaram a elaboração deste trabalho, no qual tem como objetivo a realização do estudo analítico quanto à complexidade computacional de algoritmos genéticos (AG) baseados em Lógica Fuzzy (LF) para auxiliar Sistemas de Apoio à Decisão¹ (SAD).

A complexidade trata da eficiência dos algoritmos, para Toscani e Veloso (2012, p. 14) as principais medidas de complexidade são o tempo e o espaço. Então ao se basear no tempo de execução é possível analisar e medir o comportamento do algoritmo, permitindo realizar comparações e identificar o melhor algoritmo para solucionar o problema.

Surge a motivação da implementação de algoritmos utilizando técnicas da Inteligência Artificial (IA) para auxiliar um SAD, tornando-o mais eficiente em lidar com as restrições impostas no melhor tempo de execução.

¹ Há autores, como Turban e Volonino (2013, p. 31), que denominam este tipo de sistema como Sistemas de Suporte à Decisão.

1.2 CONTEXTUALIZAÇÃO

A sociedade está em constante transformação e a cada dia novos problemas surgem aos já existentes. Para alguns desses problemas, já se conhece a solução, entretanto, para outros não. Não nos referimos a problemas inviáveis ou infactíveis², mas aos problemas que são extremamente complexos, que demonstram imprevisibilidade de comportamento, sensíveis a pequenas variações e a ausência de solução em tempo aceitável.

Diante do cenário apresentado, os sistemas de informação (SI) tornam-se ferramentas de grande importância na abordagem para a resolução de problemas factíveis. Conforme a definição abaixo:

Um **sistema de informação (SI)** pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. Além de dar apoio à tomada de decisão, à coordenação e ao controle, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos e criar novos produtos. (LAUDON e LAUDON, 2011, p. 12)

Um SI é classificado de acordo com o seu foco de atuação e por suas características. Estes sistemas oferecem soluções para problemas provenientes do ambiente organizacional, pois são programas constituídos por uma série de regras (algoritmos clássicos) que exploram os dados do espaço de busca e os processam até a obtenção de um resultado adequado.

Muitos resultados obtidos na resolução de problemas de grande complexidade não são soluções ótimas³, devido a uma combinação de fatores que degradam o desempenho do sistema. A quantidade de memória, o tempo necessário de processamento e a complexidade dos algoritmos como os principais fatores associados para definir um problema como intratável. Para Linden (Algoritmos Genéticos em Java, 2013, p. 52), problemas intratáveis são: “[...] aqueles cujo tempo necessário para resolvê-lo é considerado inaceitável para os requerimentos do usuário da solução”.

² São os problemas em que a solução viola pelo menos uma das restrições exigidas.

³ Solução que atende todos os critérios para alcançar o objetivo de forma otimizada.

Como memórias e processadores são componentes internos que podem variar em diversas características de máquina para máquina, são influências que devem ser desconsideradas no cálculo do tempo de execução. De acordo com Linden (2008, p. 8) se mudarmos o ambiente, as constantes serão alteradas, ao eliminar as constantes a taxa de crescimento não será alterada.

O tempo de execução de um algoritmo é descrito através da notação O. A definição por Linden (2008, p. 9), “Esta notação define os tempos de execução em termos de funções fundamentais, eliminando constantes e outras funções”. Desta forma é possível analisar a complexidade de um algoritmo, permitindo distinguir qual algoritmo é mais rápido na execução de uma determinada funcionalidade.

Com base na notação O, é possível verificar que os algoritmos clássicos apresentam grande dificuldade em resolver problemas complexos ou em explorar um gigante espaço de busca de forma eficaz. Mediante a esta zona intratável, a IA vem sendo utilizada para substituir a utilização das abordagens clássicas, de forma a introduzir maneiras de simular o raciocínio e a aprendizagem humana em funções computacionais, propor novos métodos de busca, compreender ideias e linguagens.

Para Turban, Wetherbe e Mclean (2007, p. 430-431) a computação convencional tem como base os algoritmos, contendo as regras que devem ser executadas e utilizando os dados para encontrar a solução. Enquanto que na computação com IA se baseia no processamento simbólico do conhecimento, que consiste em representar objetos e processos por meio de símbolos, manipulando-os com o objetivo de orientar ou fazer recomendações para a solução de problemas.

Neste trabalho será estudada uma técnica relevante para a IA, os AG, que compreende a aplicação de técnicas de busca inspiradas na evolução natural das espécies. De forma simplificada, Linden (2008, p. 44) resume e define, “Eles combinam a sobrevivência entre os melhores com uma forma estruturada de troca de informação genética entre dois indivíduos para formar uma estrutura heurística de busca”.

Outra técnica da IA que será estudada neste trabalho, o uso da LF, que permite representar graus de certeza mais próximos ao do raciocínio humano. De acordo com Weber (2003, p. 19-20), “Expressa exatamente os valores com que lida, permitindo representar graus de certeza, de associação ou valores de pertinência, intermediários entre os valores de verdadeiro e falso [...]”.

Segundo Baltzan e Phillips (2012, p. 39) os sistemas que passaram a utilizar de recursos de IA no lugar da computação tradicional obtiveram crescimento radical na velocidade e na consistência da tomada de decisão, da solução de problemas com informações incompletas e da resolução de questões complexas.

Enfim, mediante a afirmação acima, pressupõe que um SAD ao se utilizar da Gestão de Conhecimento (GC), dos AG e da LF, poderá se tornar mais eficiente e especializado em resolver problemas que possuam elevado grau de dificuldade e realizar buscas precisas no melhor tempo possível.

Para verificar se os recursos da IA realmente tornaram um sistema mais eficiente, será utilizada como referência a literatura especializada e os diversos exemplos para realizar comparações e buscar uma conclusão sobre o tema proposto por este trabalho.

1.3 METODOLOGIA

A construção do presente trabalho se estrutura na pesquisa bibliográfica que abrange o tema abordado, publicada em livros, artigos, periódicos e entre outras fontes. Com o embasamento teórico desta pesquisa será possível realizar a implementação dos algoritmos utilizando tecnologia Java.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em sete capítulos, dispostos da seguinte forma:

No Capítulo 2 é introduzido o assunto sobre SI, SAD, SBC, apresentando suas definições, suas características e demais detalhes relacionados ao assunto.

No Capítulo 3 é introduzido o assunto sobre AG, apresentando a teoria, o algoritmo básico, a estrutura geral desta técnica e demais detalhes pertinentes a este assunto.

No Capítulo 4 é introduzido o assunto sobre LF, apresentando a sua definição, suas características, um exemplo de sua aplicação nos SAD e demais detalhes relacionados ao assunto.

No Capítulo 5 são apresentados os algoritmos em Java, bem como, o estudo de sua eficácia em um caso de uso da aplicabilidade dos AG para auxiliar um sistema SAD.

Finalmente, no Capítulo 6, são apresentadas as conclusões obtidas e sugestões para a realização de trabalhos futuros.

2 SISTEMAS DE INFORMAÇÃO

Neste capítulo são apresentadas as referências teóricas relacionadas ao SI, destacando conceitos, objetivo, tipologia e características. Entretanto, o estudo será concentrado em um tipo de SI, o SAD, base para a elaboração deste trabalho.

2.1 CONCEITOS E DEFINIÇÕES

Um SI é definido como um conjunto de elementos que operam de forma integrada para reunir, buscar, processar, armazenar e distribuir informações para auxiliar no apoio de tomada de decisões, a coordenação e o controle de uma organização. (LAUDON e LAUDON, 2011, p. 12).

Para Rezende (2005, p. 26-27), a informação proporcionada por um SI é formada por “todo o dado *trabalhado, útil, tratado*, com valor significativo ou agregado a ele e com um sentido natural e lógico [...]”.

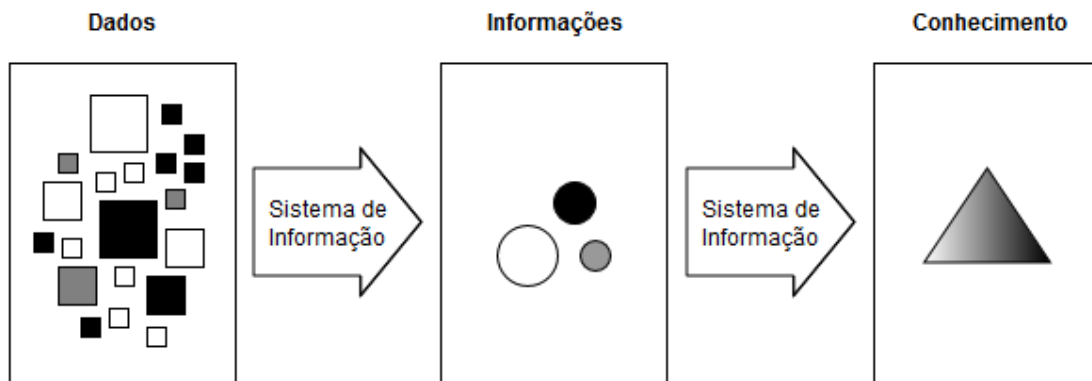


Figura 1: Transformação dos dados pelos Sistemas de Informação.

A Figura 1 apresenta a relação dos dados e os SI. De acordo com Turban, Wetherbe e Mclean (2007, p. 63), os conceitos referentes a dados, informação e conhecimento podem ser definidos conforme a Tabela 1:

Tabela 1: Dados, Informação e Conhecimento.

Tipo	Características
Dados	<ul style="list-style-type: none"> - São itens referentes a uma descrição primária de objetos, eventos, atividades e transações. - Isolados não transmitem significado específico. - Podem ser numéricos, alfanuméricos, figuras, sons ou imagens. - São armazenados e organizados em bancos de dados.
Informação	<ul style="list-style-type: none"> - É o conjunto de dados organizados de forma a possuir sentido e valor. - Requer unidade de análise para interpretação, exigindo consenso em relação ao significado com a mediação humana. - Seu uso é mais específico e maior valor agregado em relação aos dados em sua simplicidade.
Conhecimento	<ul style="list-style-type: none"> - Consiste de dados e informações organizados e processados para transmitir compreensão, experiência, aprendizado acumulado e técnica, quando se aplicam a determinado problema ou atividade. - Informação valiosa de difícil captura, transferência e estruturação. - Fornece conhecimento organizacional de alto valor potencial.

Segundo O'Brien e Marakas (2007, p. 24), o SI para gerar o seu produto, a informação, realiza sempre três ações: a entrada, o processamento e a saída (resultado). A Figura 2 apresenta as atividades básicas de um SI.

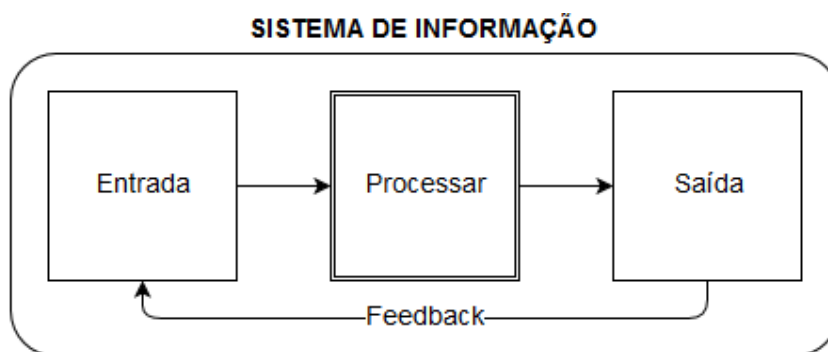


Figura 2: Atividades básicas de um Sistema de Informação.

Fonte: Adaptado de Laudon e Laudon (2011, p. 13).

Para Laudon e Laudon (2011, p. 12) e O'Brien e Marakas (2007, p. 27), a entrada compreende na captura ou coleta de dados brutos, o processamento converte esses dados em uma forma mais significativa e a saída disponibiliza as informações processadas para serem utilizadas. Os SI também requerem um *feedback*⁴ para ajustar os dados de entrada.

⁴ Retroinformação ou Alimentação de Retorno, que consiste na ação de realizar comentários ou informações sobre uma atividade que foi realizada com o objetivo de avaliação.

Basicamente, é perceptível o objetivo intrínseco de um SI, que é o de realizar o processamento dos dados e retornar com uma informação inteligível ou o de obter conhecimento. Entretanto, para perceber suas diferenças e seu grau de especialização é necessário compreender o tipo de problema que estão propostos a resolver, os elementos que compõem sua estrutura, o nível organizacional e área que proporcionam suporte.

2.2 RESOLUÇÃO DE PROBLEMAS

Atualmente, muitas organizações são apoiadas por SI para resolver diversos tipos de problemas no seu meio organizacional. Segundo Turban, Wetherbe e Mclean (2007, p. 62), “Os sistemas de informação são elaborados para facilitar a concretização de determinados objetivos”, ou seja, o objetivo é a solução para um determinado problema (2007, p. 40).

Para Laudon e Laudon (2011, p. 19), os problemas enfrentados pelas organizações envolvem uma série de fatores, que podem ser agrupados em três categorias: organização, tecnologia e pessoas.

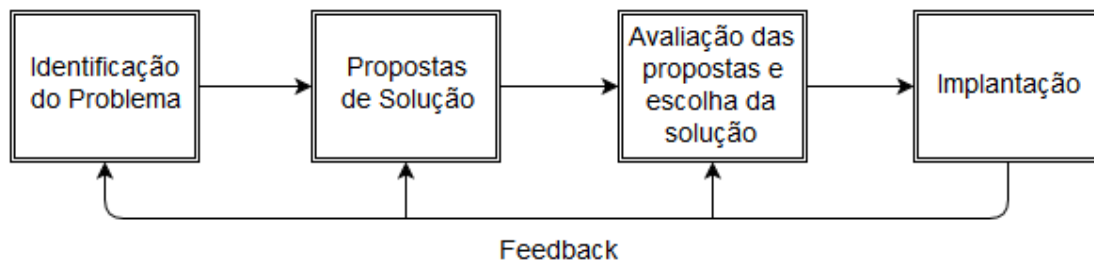


Figura 3: Processo de resolução de problemas.

Fonte: Adaptado de Laudon e Laudon (2011, p. 19).

Existe um modelo simples para o processo de resolução de problemas organizacionais por meio de SI. Na Figura 3, são apresentados os quatro passos desse modelo. A definição para os quatro passos que compõe o processo de resolução de problemas segundo definição de Laudon e Laudon (2011, p. 19-21):

- 1) A identificação do problema, como o próprio nome indica, deve ser identificada se a existência do problema é concreta, qual o problema, quais suas causas, o que pode ser feito e dados sobre os recursos disponíveis para viabilizar sua resolução.
- 2) Propostas de solução, neste passo devem ser levantados o maior número de soluções possíveis, que abrangem as três categorias: organização, tecnologia e pessoas.
- 3) Na avaliação e escolha, é necessário considerar todas as propostas de solução e verificar qual delas é a mais benéfica, ou a “melhor” solução. Neste ponto, devem ser levados em consideração as atitudes e o apoio das pessoas envolvidas, pois uma solução que requer o apoio destes e não o tem se torna um grande fracasso.
- 4) A implantação consiste em pegar a solução selecionada que envolva SI, adquirir ou desenvolver essa solução e introduzi-la na organização.

Com os quatro passos, é possível descobrir a causa e alcançar uma solução. Entretanto, a implantação de uma solução é um processo muito complexo e será necessário alcançar outras áreas para buscar essa solução.

Esta análise de resolução de problemas é muito importante, pois permite que sejam evitados novos problemas. Por exemplo, ao desenvolver um sistema o foco é construir algo que atenda um objetivo específico, em consequência haverá altos custos com a equipe de execução desta tarefa e o tempo poderá ser muito grande para a obtenção da solução.

Enquanto adquirir um sistema poderá acarretar em realizar adaptações ao software para prover a solução desejada e que contemple o processo de negócio da organização. Sem uma análise previa há chances de que a solução de um problema se torne a causa de novos problemas.

2.3 CLASSIFICAÇÃO DOS SISTEMAS

De acordo com Laudon e Laudon (2011, p. 14-15; 42), uma organização está estruturada por diferentes níveis e especializações, processos de negócios,

diversos interesses e pontos de vista. Em consequência, existirá uma grande variedade de problemas. Como nenhum sistema isolado consegue fornecer todas as informações de que uma organização necessita, existem diversos tipos de sistemas.

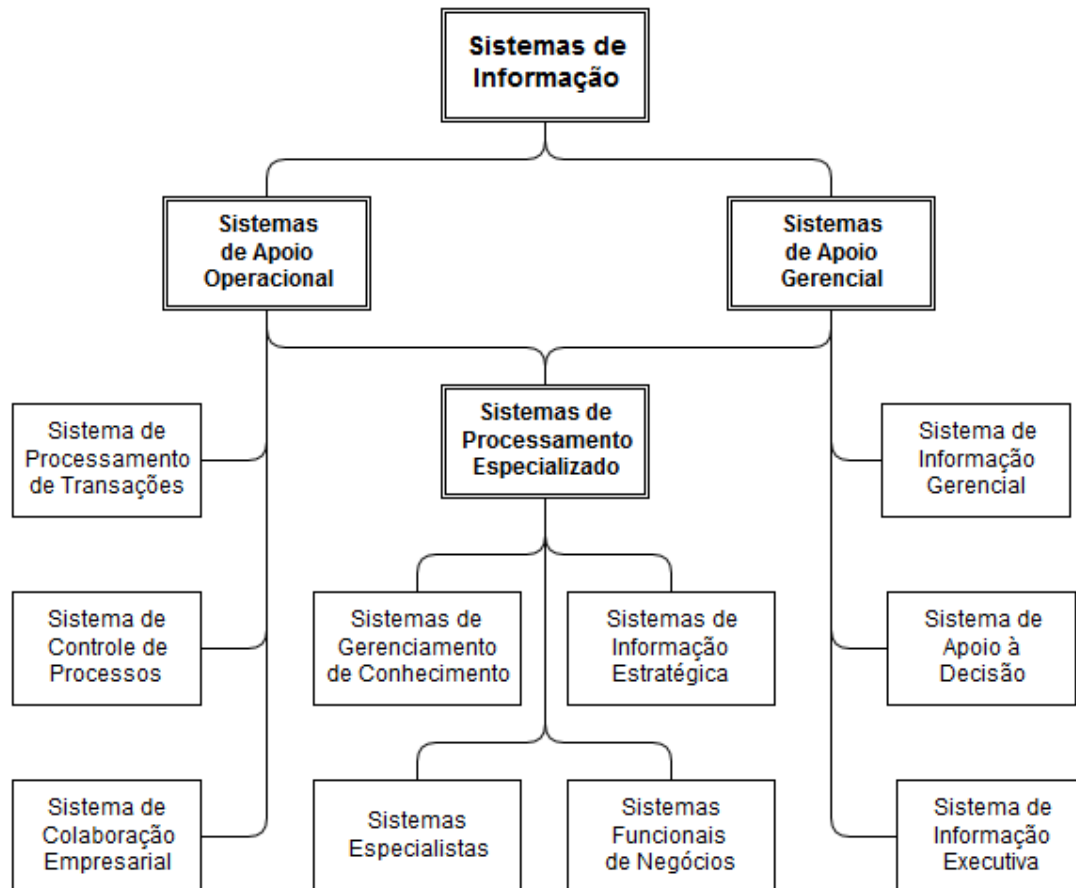


Figura 4: Classificação dos SI.

Fonte: Adaptação de O'Brien e Marakas (2007, p. 14).

Segundo Loh (2014, p. 10-11), existem 31 tipos de SI, classificados de acordo com seus objetivos e tipos de informações que manipulam. Entretanto, o objetivo desta seção não é o de apresentar cada um destes 31 SI, mas o de apresentar os principais tipos de SI, desta forma, será baseado segundo a classificação de O'Brien e Marakas (2007, p. 13) e Laudon e Laudon (2011, p. 42), no qual os SI são classificados como sistemas operacionais ou sistemas de informação gerencial, como ilustra a Figura 4.

2.3.1 Sistemas de Apoio Operacional

Também conhecidos como Sistemas de Suporte a Operações, são sistemas que processam operações e transações rotineiras, apenas controlam dados para serem armazenados e posteriormente utilizados pelos responsáveis pela tomada de decisão.

A Tabela 2 apresenta os tipos, as características e os exemplos de acordo com O'Brien e Marakas (2007, p. 13-15), dos sistemas classificados como Sistemas de Apoio Operacional:

Tabela 2: Sistemas de Apoio Operacional.

Tipo	Características
Sistemas de Processamento de Transações	Processa os dados resultantes das transações de negócios, atualiza banco de dados operacionais e produz documentos de negócios. Exemplos: Processamento de vendas e estoque e de contabilidade.
Sistemas de Controle de Processos	Monitora e controla o processo industrial. Exemplos: Refinação de petróleo, geração de energia e sistemas de produção de aço.
Sistemas de Colaboração Empresarial	Apoio à comunicação e colaboração para equipe, grupos de trabalho e empresa. Exemplos: Sistemas de e-mail, chat e videoconferência.

Fonte: Adaptação de O'Brien e Marakas (2007, p. 14).

2.3.2 Sistemas de Apoio Gerencial

São os sistemas que sintetizam os dados agrupados pelos sistemas de apoio operacional, transformando-os em informações agrupadas com a finalidade de fornecer suporte aos responsáveis pela tomada de decisão.

Segundo O'Brien e Marakas (2007, p. 15), "O fornecimento de informação e suporte para a tomada de decisão por todos os tipos de gerentes e profissionais de negócios é uma tarefa complexa", desta forma, na Tabela 3 apresenta os vários tipos de SI que fornecem suporte à vários graus de responsabilidade.

Tabela 3: Sistemas de Apoio Gerencial.

Tipo	Características
Sistemas de Informação Gerencial (SIG)	Fornecer a informação na forma de relatórios e telas pré-especificados para a tomada de decisão de negócios. Exemplos: Relatórios de análise de vendas, desempenho da produção e tendência de custo.
Sistemas de Apoio à Decisão (SAD)	Prover suporte <i>ad hoc</i> ⁵ interativo para os processos de tomada de decisão de gerentes e outros profissionais de negócios. Exemplos: Sistemas para cálculo de preço de produtos, previsão de rentabilidade e tendência de custo.
Sistemas de Informação Executiva	Fornecer informação fundamental dos SIG, SAD e outras fontes adaptadas às necessidades de informação ao grupo de executivos. Exemplos: Sistema de fácil acesso às análises do desempenho dos negócios e às ações dos concorrentes.

Fonte: Adaptação de O'Brien e Marakas (2007, p. 15).

2.3.3 Sistemas de Processamento Especializado

Os sistemas que se enquadram nesta categoria podem fornecer suporte para aplicações tanto operacionais como gerenciais, de acordo com O'Brien e Marakas (2007, p. 16). Na tabela 4 são apresentados os SI desta categoria.

Tabela 4: Sistemas de Processamento Especializado.

Tipo	Características
Sistemas Especialistas (SE)	Sistemas baseados no conhecimento que fornecem assessoria abalizada e agem como consultores técnicos a usuários. Exemplos: Conselheiro de aplicação de crédito e monitor de processos.
Sistemas de Gestão de Conhecimento	Sistemas baseados em conhecimento que suportam a criação, a organização e a disseminação do conhecimento dentro da empresa. Exemplos: Sistema de resolução de problemas de clientes.
Sistemas de Informação Estratégica	Suportam operações ou os processos de gerência sobre fornecimento de produtos e serviços estratégicos e condições para vantagem competitiva. Exemplos: Pregão eletrônico e sistemas de comércio eletrônico na web.
Sistemas Funcionais de Negócios	Suportam uma variedade de aplicações operacionais e gerenciais das funções básicas de uma companhia. Exemplos: Gerenciamento de operações e de recursos humanos.

Fonte: Adaptação de O'Brien e Marakas (2007, p. 17).

⁵ Termo utilizado para expressar um fim específico, "para isto".

2.4 EVOLUÇÃO

A utilização dos SI evoluiu ao longo dos anos, conforme surgiam novos problemas, as organizações buscaram novos sistemas baseados em computadores para resolvê-los, desta forma, foi aberta uma porta para novas possibilidades de analisar os dados históricos, entender as tendências atuais e antecipar as próximas. A Figura 5 apresenta a evolução dos problemas ao longo das décadas.

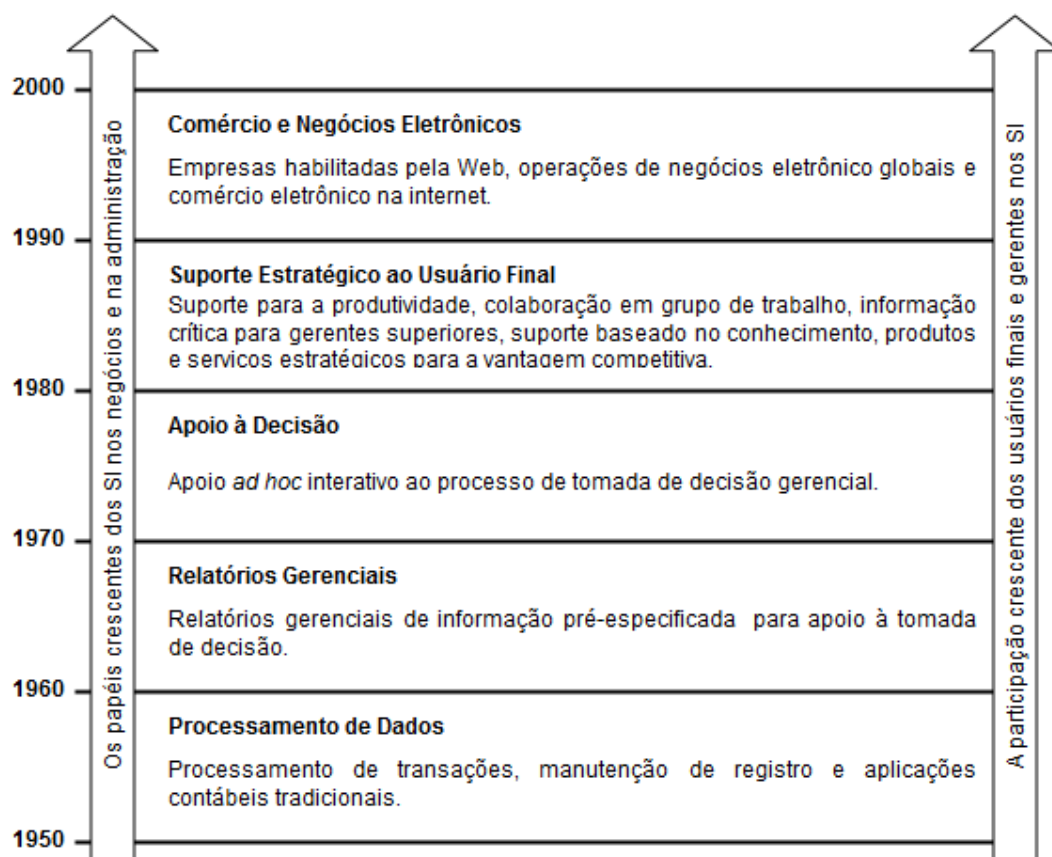


Figura 5: Evolução da solução de problemas.

Fonte: Adaptação de O'Brien e Marakas (2007, p. 11).

Segundo O'Brien e Marakas (2007, p. 10-11), a maior parte dos SI até os anos 60 eram apenas de Sistemas de Processamento de Transações, após os anos 60, foi acrescentado os SIG, que utilizavam os dados armazenados em relatórios com informações úteis na tomada de decisão. Nos anos 70, surgiram os SAD para

fornecer suporte que não eram possíveis com os SIG, que se concentrava em fornecer aos usuários administrativos finais o suporte *ad hoc* e interativo dos processos de tomada de decisão.

Continuando com O'Brien e Marakas (2007, p. 11), nos anos 80, surgiram novos papéis para os SI, com o principal foco nos usuários finais, surgindo também o conceito de Sistemas de Informação Executiva, para apoiar os altos executivos a obter informações críticas da corporação. Neste período, também foram inseridos Sistemas Especialistas, Sistemas baseados no conhecimento e avanços no desenvolvimento SI com técnicas de IA.

Finalmente, segundo O'Brien e Marakas (2007, p. 12), desde a década de 80 e seguindo pelos anos 90, o conceito do papel estratégico dos SI foi intensificado, a tecnologia da informação se torna um componente dos processos de negócio, produtos e serviços. De meados até o final dos anos 90, surgiram os revolucionários Sistemas Integrados ou ERP (*Enterprise Resource Planning* – Sistemas de Gestão Empresarial). (JUNIOR, 2008). Nos anos 90, com o rápido crescimento da internet, as empresas passaram usar a Web, criando sistemas de comércio e negócios eletrônicos globais.

2.5 SISTEMA DE APOIO À DECISÃO

Começaremos esta seção com a definição de SAD:

O **sistema de apoio à decisão (SAD)** é um sistema de informação baseado em computador que combina modelos e dados, em uma tentativa de solucionar problemas semi-estruturados com grande envolvimento por parte do usuário. (TURBAN, WETHERBE e MCLEAN, 2007, p. 370)

Os SAD surgiram devido à competitividade das organizações, com a presença de problemas gerados com a carência de interpretação das informações pré-especificadas geradas pelos SIG, muitos dados armazenados sem utilidade que apresentassem algum sentido, tendências e experiências.

Com o surgimento dos SAD, ocorreu uma significativa mudança no comportamento das corporações, que passaram a utilizar o sistema para auxiliar nas decisões gerenciais exclusivas e complexas, através de simulações de problemas específicos do mundo real.

Razões para o crescimento do uso dos SAD:

Muitas empresas estão utilizando o SAD para melhorar o processo decisório. As razões citadas pelos gerentes são:

- Necessidade de informações novas e mais precisas.
- Necessidade de ter informações mais rapidamente.
- O monitoramento das inúmeras operações de negócio da empresa estava cada vez mais difícil.
- A empresa estava operando em uma economia instável.
- A empresa enfrentava maior concorrência nos mercados interno e externo.
- Os sistemas instalados na empresa não apoiavam adequadamente os objetivos de maior eficiência, rentabilidade e ingresso em mercados lucrativos.
- O departamento de sistemas de informação não conseguia mais atender à diversidade de necessidades imediatas da empresa e de seus executivos e não havia funções de análise de negócio embutidas nos sistemas existentes.

(TURBAN, WETHERBE e MCLEAN, 2007, p. 371)

2.5.1 Características e Capacidades

Para Turban, Wetherbe e Mclean (2007, p. 371), não existe um consenso sobre as características e capacidades que constituem um SAD. A maioria dos SAD possuem somente alguns dos atributos apresentados abaixo:

1. O SAD dá apoio aos tomadores de decisão em todos os níveis gerenciais, seja individualmente seja em grupos, principalmente em situações semi-estruturadas e não-estruturadas, combinando a capacidade humana de julgamento com a informação objetiva.
2. O SAD dá apoio a diversas decisões interdependentes e/ou sequenciais.
3. O SAD dá apoio a todas as fases do processo decisório – inteligência, desenho, escolha e implementação – bem como a uma variedade de processos e estilos de decisão.
4. Com o tempo, o SAD é adaptado pelo usuário para poder lidar com mudanças de condições.
5. O SAD é fácil de construir e usar em muitos casos.
6. O SAD promove o aprendizado, o que conduz a novas demandas e ao aprimoramento do aplicativo, levando por sua vez a novo aprendizado e assim por diante.
7. O SAD geralmente utiliza modelos quantitativos (padrão e/ou sob medida).
8. O SAD mais avançado é equipado com um componente de gestão do conhecimento que permite a solução eficiente e eficaz de problemas bastante complexos.
9. O SAD pode ser disseminado por meio da Web.
10. O SAD permite a realização de *análises de sensibilidade*.

(TURBAN, WETHERBE e MCLEAN, 2007, p. 371)

A análise de sensibilidade pode ser descrita como o processo que permite realizar simulações para observar o comportamento do modelo e o impacto dos resultados obtidos de acordo com a alteração dos dados de entrada do problema. Conforme Turban, Wetherbe e Mclean (2007, p. 371), “A análise de sensibilidade é extremamente valiosa no SAD, pois torna o sistema flexível e adaptável a condições mutáveis e às inúmeras exigências das diferentes situações de decisão”.

Existem dois tipos de análises de sensibilidade segundo Turban, Wetherbe e Mclean (2007, p. 372):

- a) **Análise de hipóteses:** Neste modelo é possível realizar previsões e premissas baseadas nos dados de entrada. Os resultados dependem dessas premissas para avaliar o impacto das alterações sobre a solução proposta.
- b) **Atingimento de meta:** Neste modelo, a análise procura descobrir os valores de entrada necessários para obter um resultado desejado, de forma inversa à análise de hipóteses.

2.5.2 Estrutura e Componentes

A Figura 6 apresenta de acordo com Turban, Wetherbe e Mclean (2007, p. 372-373), todos os SAD consistem dos seguintes componentes básicos:

- **Subsistema de Gestão de Dados:** Semelhante a qualquer outro sistema de dados, entretanto, este é exclusivamente utilizado pelo SAD. Concentra coleções de dados atuais e históricos extraídos de diversas fontes, sendo inseridos nos modelos do SAD conforme a necessidade, organizados como um banco de dados relacional.
- **Subsistema de Gestão de Modelo:** Contêm modelos que fornecem recursos analíticos do sistema e uma ferramenta construtora de modelos para as aplicações de SAD. Este subsistema possui outros elementos: base de modelos, sistema de gestão de base de modelos, guia de modelos e execução, integração e comando de modelos.

- **Subsistema de Interface com o Usuário:** É o canal de interação e comunicação entre usuário e SAD. Considerado o componente mais importante, devido à premissa de prover uma interface amigável ao tomador de decisão, de forma a ser intuitiva, de fácil manuseio e rápido aprendizado. Pode ser gerenciado por um sistema de gerenciamento de interface de usuário.

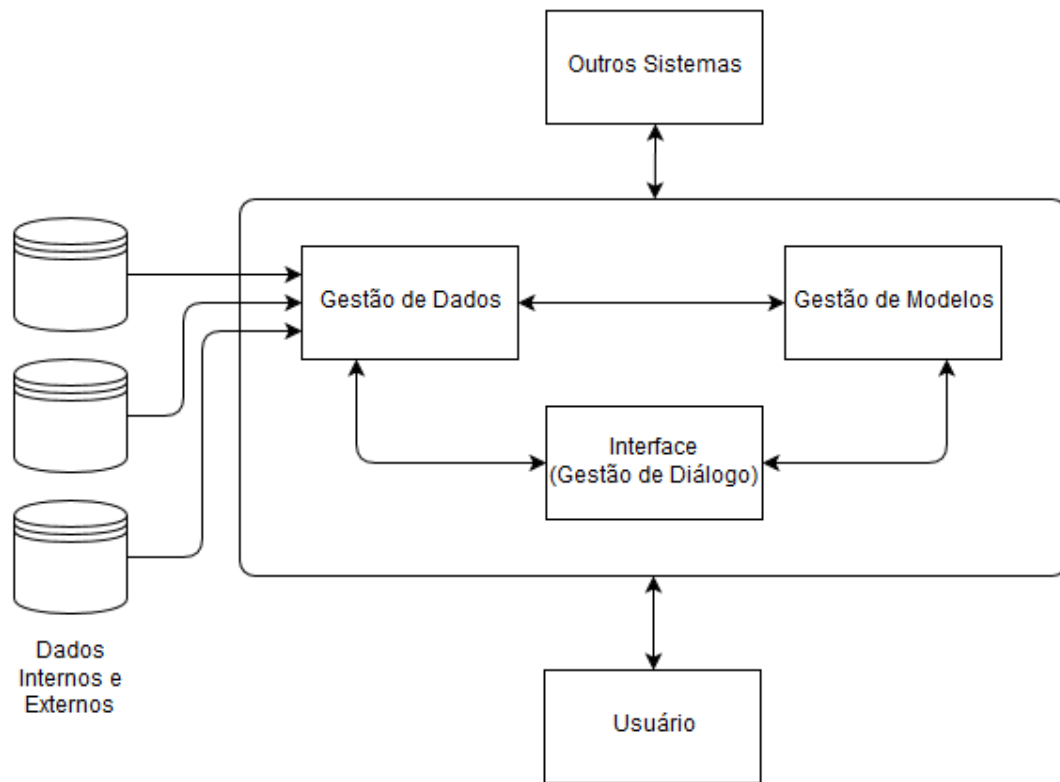


Figura 6: Estrutura de um SAD.

Fonte: Adaptação de Turban, Wetherbe e Mclean (2007, p. 374).

2.5.3 Categorias Especiais

Existem também categorias especiais de SAD que possuem foco quanto ao nível de apoio à decisão. De acordo com Hackathorn e Keen (1981) citado por Turban, Wetherbe e Mclean (2007, p. 378), existem três níveis distintos de apoio à decisão: individual, em grupo e na empresa.

O apoio em grupo é realizado pelo Sistema de Apoio à Decisão em Grupo (SADG). Laudon e Laudon (2011, p. 335) o descreve como “um sistema interativo, baseado em computador, para facilitar a resolução de problemas não estruturados, por um conjunto de profissionais que tomam decisões trabalhando juntos como um grupo, no mesmo lugar ou não”.

Os SADG fornecem ferramentas tecnológicas moldadas especificamente para a tomada de decisão em grupo através de reuniões. De acordo com Turban, Wetherbe e Mclean (2007, p. 375), o SADG conta com uma sala de decisão para a realização de reuniões com suporte a videoconferência, a utilização de *groupware*⁶ e software para gerar e organizar ideias, priorização e formulação de políticas.

O objetivo do SADG é melhorar a produtividade das reuniões de decisão, entretanto, a eficiência depende da natureza do problema e das decisões resultantes para sua solução, do grupo organizado para o processo decisório e da forma como a reunião é planejada e conduzida.

Quando o foco de apoio é no nível da empresa⁷, é fornecido pelo Sistema de Apoio à Decisão Empresarial (SADE), como Hackathorn e Keen (1981) citado por Turban, Wetherbe e Mclean (2007, p. 378) definem, “sistema que focaliza uma tarefa ou atividade da empresa envolvendo uma sequência de operações e de tomadores de decisão”.

Uma importante característica atribuída aos SADE é o recurso *drill down*, para Laudon e Laudon (2011, p. 335), este recurso “permite passar do nível de dados resumidos para níveis cada vez mais detalhados”, Turban, Wetherbe e Mclean (2007, p. 378) acrescentam que este recurso “permite aos usuários, obter detalhes, e detalhes de detalhes, de qualquer informação desejada”.

Segundo Laudon e Laudon (2011, p. 335), os sistemas SADE ajudam a controlar o desempenho da empresa, acompanhar os processos de concorrência, auxilia na identificação de alterações nas condições do mercado, detectar problemas e oportunidades, bem como prever tendências.

Atualmente, as organizações modernas têm optado por aplicações que operam dentro de uma plataforma global baseada na Web. De acordo com Turban,

⁶ De acordo com Rainer e Cegielski (2012, p. 166), são produtos de software que aceitam grupos de pessoas que colaboram uns com os outros em uma tarefa ou objetivo comum e proporcionam um modo de os grupos compartilharem recursos.

⁷ Para Rainer e Cegielski (2012, p. 316), atribuem este nível como organizacional, em consequência, esta categoria é denominada de Sistema de Apoio à Decisão Organizacional.

Wetherbe e Mclean (2007, p. 381), “a Web é um meio perfeito para instalar recursos de apoio à decisão de uma maneira global”. O SAD baseado na Web pode suportar qualquer nível de apoio à decisão.

2.5.4 Modelagem e Modelos

Os modelos exercem um papel muito importante na modelagem realizada pelo SAD, pois são representações simplificadas ou abstrações da realidade. Com a modelagem é possível realizar simulações, análises e experiências sobre um modelo baseado do mundo real.

Os benefícios da modelagem são:

1. O custo da experimentação virtual é muito baixo do que os custos da experimentação conduzida com um sistema real.
2. Os modelos permitem fazer uma compactação simulada do tempo. Anos e anos de operação podem ser simulados no computador em poucos segundos.
3. É muito mais fácil manipular o modelo, mudando as variáveis, do que manipular o sistema real. A experimentação é, portanto, mais fácil e não interfere nas operações diárias da empresa.
4. O custo de erros cometidos durante uma verdadeira experiência de tentativa e erro é muito mais baixo quando se usam modelos para teste virtual.
5. O ambiente atual possui uma grande dose de incerteza. A modelagem permite ao gerente lidar melhor com essa incerteza e calcular os riscos envolvidos em ações específicas.
6. Os modelos matemáticos permitem a análise de um número gigantesco e quase infinito de possíveis soluções alternativas. Com tecnologia e comunicações tão avançadas hoje em dia, os gerentes frequentemente contam com um grande número de alternativas à sua escolha.
7. Os modelos melhoram e reforçam o aprendizado e dão apoio ao treinamento. (TURBAN, WETHERBE e MCLEAN, 2007, p. 364)

De acordo com Turban, Wetherbe e Mclean (2007, p. 365), os modelos podem representar sistemas ou problemas com vários graus de abstração. Eles são classificados em quatro grupos:

- **Modelo baseado em ícone (de escala):** São os de menor grau de abstração, réplicas físicas de um sistema, baseados geralmente em uma escala diferente da original e podem ser apresentados em escala tridimensional.

- **Modelo análogo:** Diferente do modelo baseado em ícone, não se parece com o sistema real, apenas possui o seu comportamento, ou seja, uma representação simbólica do real.
- **Modelo matemático (quantitativo):** A maior parte da análise do SAD é numérica, não podendo ser representados na forma icônica ou análoga, devem ser utilizados modelos matemáticos.
- **Modelos mentais:** Representação interna e conceitual usada para gerar descrições da estrutura de um problema de acordo com a interpretação das pessoas. Este modelo inclui crenças, premissas, relações e fluxos de trabalho da forma que são percebidos pelo indivíduo.

2.6 SISTEMAS BASEADOS EM CONHECIMENTO

A utilização dos Sistemas Baseados em Conhecimento (SBC) começou a partir dos anos 80, com presença significativa na resolução de problemas complexos e quando da existência da necessidade de algum conhecimento específico para resolvê-lo. Definição de SBC:

Sistemas Baseados em Conhecimento são programas de computador que usam o conhecimento representado explicitamente para resolver problemas. Eles manipulam conhecimento e informação de forma inteligente e são desenvolvidos para serem usados em problemas que requerem uma quantidade considerável de conhecimento humano e de especializações. (REZENDE, PUGLIESI e VAREJÃO, 2005, p. 15)

A importância dos SBC para diversas organizações esta na capacidade de preservar, aproveitar e fazer uso da experiência adquirida pelos envolvidos no processo de tomada de decisão. Este conhecimento torna-se permanentemente acessível, facilmente recuperável e pode ser amplamente utilizado por todos dentro da organização.

2.6.1 Diferente dos Demais

Existem muitas diferenças entre os SBC aos demais SI convencionais. Segundo Rezende, Pugliesi e Varejão (2005, p. 17-18) os SBC diferem dos sistemas convencionais quanto a sua organização, da forma que incorporam conhecimento, em como executam e a impressão que causam aos usuários com quais interagem. A Tabela 5 apresenta as principais diferenças entre SI Convencionais dos SBC.

Tabela 5: Principais diferenças entre SI Convencionais e SBC.

SI Convencionais	SBC
Os dados são organizados por estruturas de dados, as quais estão relacionadas com a arquitetura do computador.	Os dados são organizados por meio da representação do conhecimento, que abstrai completamente os detalhes da máquina onde o conhecimento é processado.
Estruturas de dados apenas representam os dados e suas relações.	Representações do conhecimento explicitam os conceitos do domínio do problema, suas relações e as regras de dedução.
Empregam algoritmos determinísticos para realizar suas funções de busca.	Realizam busca em um espaço de possíveis soluções através de métodos de resolução de problemas e fazem uso intensivo de conhecimento heurístico para tornar a busca efetiva.
O conhecimento do domínio e o método de resolução de problemas estão embutidos no código do programa, tornando sua inspeção, seu entendimento e sua manutenção um processo oneroso e complicado.	O conhecimento é representado de forma evidente e separado do programa que o manipula e interpreta, tornando assim mais fácil o entendimento do conhecimento usado para a resolução do problema.
A explicação do raciocínio é muito difícil e costuma não ser realizado.	A explicação do raciocínio é parte essencial, fácil e sempre realizado.

Fonte: Adaptação de Rezende, Pugliesi e Varejão (2005, p. 18-19).

É importante também diferenciar os SBC dos SE, pois ambos trabalham com o conhecimento. De acordo com Rezende, Pugliesi e Varejão (2005, p. 19), os SBC são sistemas com a capacidade de resolver problemas usando conhecimento específico sobre um domínio da aplicação. Enquanto que os SE são SBC que resolvem problemas ordinariamente resolvidos por um especialista humano, por isso requerem conhecimento sobre a habilidade, a experiência e as heurísticas⁸ usadas pelo especialista.

⁸ São algoritmos polinomiais que não têm garantia nenhuma sobre a qualidade da solução obtida, mas tendem a encontrar a solução ótima ou ficar bem próximos dela. (LINDEN, 2008, p. 41).

Os SBC podem ser classificados como SE quando o propósito do mesmo for voltado para aplicações nas quais o conhecimento a ser manipulado se restringe a um domínio específico e com alto grau de especialização.

2.6.2 Estrutura e Componentes

De acordo com Rezende, Pugliesi e Varejão (2005, p. 22-29), nem todos os SBC apresentam uma mesma estrutura, entretanto, a maioria apresenta estrutura semelhante.

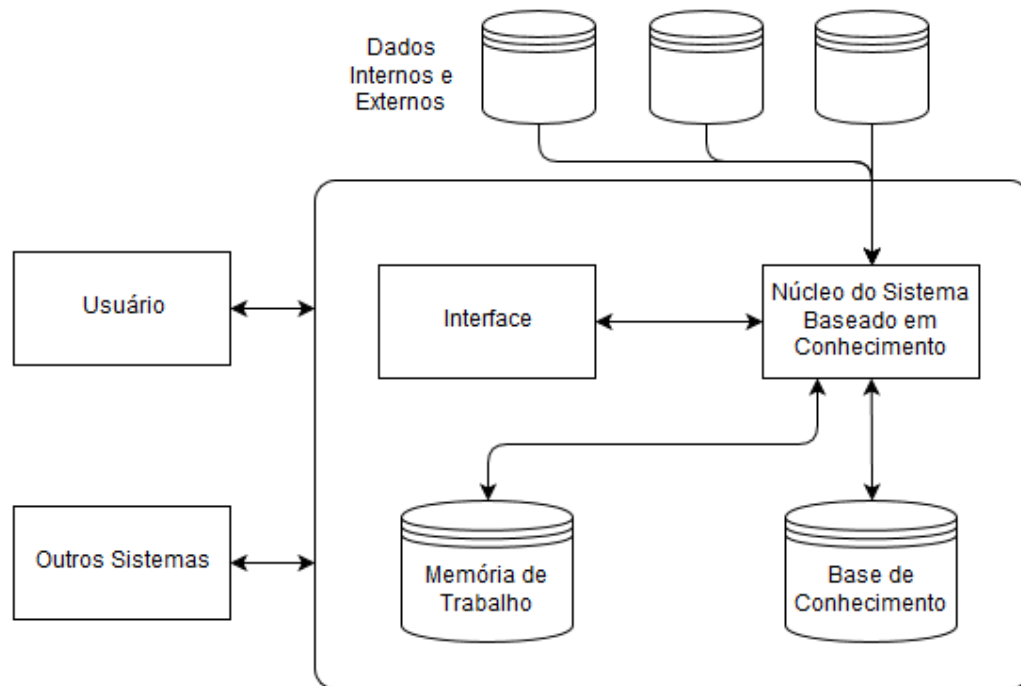


Figura 7: Estrutura de um SBC.

Fonte: Adaptação de Rezende, Pugliesi e Varejão (2005, p. 23).

A Figura 7 apresenta a estrutura geral e básica de um SBC é constituída pelos seguintes componentes:

- **Núcleo do Sistema Baseado em Conhecimento:** Denominado também por *Shell*⁹. É o componente mais importante do sistema, responsável em desempenhar as principais funções e gerir o mecanismo de inferência.
- **Base de conhecimento:** Armazena todo o conhecimento sobre um determinado domínio, representado em várias linguagens de Representação de Conhecimento (RC) como: regras de produção, redes semânticas, *frames*¹⁰ e lógica.
- **Memória de Trabalho:** Representa a área de trabalho de um SBC, onde são registradas todas as respostas fornecidas pelo usuário ao longo das interações realizadas pelo sistema, evitando assim a perguntas repetidas ou redundantes. Além de armazenar as condições iniciais, conclusões, decisões intermediárias e até as soluções finais.
- **Subsistema de Interface com o Usuário:** É o canal de interação e comunicação entre o usuário e o SBC em ambas as direções e realizando a intermediação da representação interna do sistema e a representação mental do usuário.

2.6.3 Núcleo do Sistema

O núcleo do SBC, Responsável pelas principais funções do SBC: Controle da interação com o usuário ou com equipamentos externos; Processamento do conhecimento utilizando alguma linha de raciocínio; Justificativa ou explicação das conclusões obtidas a partir do raciocínio.

De acordo com Rezende, Pugliesi e Varejão (2005, p. 23-25), o núcleo é composto basicamente por três módulos interdependentes:

⁹ Termo que faz referência ao software quando a sua base de conhecimento se encontra vazia.

¹⁰ Segundo Malda (1987) e Minsky (1975), citado por Rezende, Pugliesi e Varejão (2005, p. 33), “É um termo usado para designar um agrupamento de conhecimentos relevantes a uma coisa, um indivíduo, uma situação ou um conceito”.

- **Módulo Coletor de Dados:** Responsável pela interação com o usuário, obtendo informações do problema em análise, através da formulação de sucessivas perguntas. Quando ativado, realiza as perguntas necessárias e a validação das respostas de acordo com funções críticas preestabelecidas.
- **Motor de Inferência:** É responsável pelo desenvolvimento do raciocínio baseado nas informações obtidas pelo Coletor de Dados e no conhecimento representado na Base de conhecimento. Este módulo processa a linguagem de representação usada na Base de Conhecimento, gerando e percorrendo espaço de busca sempre que necessário.
- **Módulo de Explicações:** É responsável pela explicação, ou justificativas, das conclusões obtidas e dos motivos pelos quais são realizadas determinadas perguntas. Os principais tipos de explicações são: “por que”, “como”, “o que acontece se” e “por que não”. É observada que a explicação fornecida ao usuário é uma interpretação de alto nível que depende do formalismo com o qual é representado o conhecimento.

De acordo com Rezende, Pugliesi e Varejão (2005, p. 25), enquanto o núcleo executa suas funções, interage com a Memória de Trabalho, onde são gravados e acessados os dados temporários, realiza também a manipulação da Base de Conhecimento, que contém representado todo o conhecimento.

2.6.4 Conhecimento e sua Representação

A estruturação do conhecimento permite a resolução eficaz e eficiente de problemas, do aprendizado dinâmico, do planejamento estratégico e na tomada de decisão, essas características tornam o conhecimento um elemento essencial e tão importante. (TURBAN, WETHERBE e MCLEAN, 2007, p. 326).

Para realizar o processo de estruturação do conhecimento, é necessário compreender sob quais formatos o conhecimento pode ser encontrado. De acordo com Cruz (2007, p. 39-41), basicamente existem dois tipos de conhecimento:

- **Conhecimento Tácito:** É todo conhecimento acumulado, fruto do aprendizado, da educação, da especialização, da cultura, *insights*¹¹, perspicácia, *know-how*¹², segredos de negócios, a compreensão e por toda a experiência adquirida em sua existência. Da mesma forma que ocorre com uma pessoa, também ocorre com uma organização. Um dos grandes desafios em qualquer organização é o de coletar, organizar e utilizar esse tipo de conhecimento.
- **Conhecimento Explícito:** É o conhecimento formal e sistemático mais simples de ser exteriorizado e documentado em linguagem formal para ser transmitido ou distribuído rapidamente e em grande escala. Este conhecimento está presente em algum tipo de material físico como em livros, manuais, relatórios, vídeos, desenhos, enfim qualquer objeto que permita desenvolver as habilidades e agregar mais conhecimento ao indivíduo.

O conhecimento, tácito ou explícito, sobre um determinado domínio deve ser representado de modo que seja compreensível ao ser humano e o computador possa processá-lo. A definição de RC por Rezende, Pugliesi e Varejão (2005, p. 29), “pode ser entendida como uma forma sistemática de estruturar e codificar o que se sabe sobre uma determinada aplicação”.

A RC deve apresentar as seguintes características:

- Ser compreensível ao ser humano, pois caso seja necessário avaliar o estado de conhecimento do sistema, a Representação do Conhecimento deve permitir a sua interpretação;
- Abstrair-se dos detalhes de como funciona internamente o processador de conhecimento que o interpretará;
- Ser robusta, isto é, permitir sua utilização mesmo que não aborde todas as situações possíveis;
- Ser generalizável, ao contrário do conhecimento em si que é individual. Uma representação necessita de vários pontos de vista

¹¹ Palavra que expressa uma percepção mental explícita, geralmente intuitiva, quando ocorre uma compreensão ou entendimento súbito que levam a resolução de um problema.

¹² Expressão inglesa que referencia o conjunto de conhecimentos, técnicas e competências adquiridas através da experiência sobre como fazer alguma coisa.

do mesmo conhecimento, de modo que possa ser atribuída a diversas situações e interpretações.

(REZENDE, PUGLIESI e VAREJÃO, 2005, p. 29-30)

De forma resumida a seguir, são apresentadas algumas técnicas mais utilizadas de RC, de acordo com Rezende, Pugliesi e Varejão (2005, p. 30-36):

- **Representação Lógica:** A lógica matemática é uma linguagem formal, que possui várias regras sintáticas de dedução, são formas de realizar inferências dedutivas a partir do formato sintático das expressões da linguagem, não levando em consideração ideias extras ou intuitivas. O termo dedução automática refere-se ao comportamento de qualquer programa de computador que realiza inferências dedutivas pelo princípio das leis da lógica matemática. Existem várias extensões da lógica matemática dentro da área de dedução automática, como lógicas não-monotônicas, temporais, descritivas e modais. Estas extensões tornam o uso da lógica mais adequada e fácil em contextos especiais.
- **Regras de Produção:** Baseados na ideia de modelar o processo decisório humano no qual se utilizam regras estruturadas do tipo: *SE condições ENTÃO conclusões FAÇA ações*. Tais regras podem expressar relacionamento lógico e equivalências de definições para simular o raciocínio humano. Na programação baseada em regras, usa-se o motor de inferência para manipular regras e fatos para produzir novos fatos até chegar a um resultado final. As regras são constituídas de forma natural para representar o conhecimento de um especialista humano.
- **Redes Semânticas:** É um grafo específico, rotulado e direcionado, os objetos (indivíduos, conceitos, situações dentro de um domínio) são representados por nós e os relacionamentos entre os objetos são representados por arestas, rotuladas com o nome da relação. Um objeto ao apresentar complexidade, pode ser decomposto em objetos mais simples. Uma das propriedades mais importantes dessas relações é a transitividade, pois permite uma declaração concisa de propriedades nos objetos mais gerais. Mecanismos de inferência podem ser utilizados para derivar essas propriedades

aos objetos mais específicos, este processo é denominado como Herança de Propriedade.

- **Frames:** É uma estrutura de dados complexa que proporciona um modo útil de modelar objetos do mundo real. *Frame* possui nome que identifica o conceito por ele definido e consiste de um conjunto de atributos, chamado de *slots*. Um *slot* possui um nome e consiste de um conjunto de atributos, denominadas de *facet*s. As *facet*s contêm informações que descrevem os *slots*, definindo os valores que os *slots* podem assumir ou descrevendo a forma de calcular ou deduzir o seu valor. Como também visto nas redes semânticas, os *frames* também possuem Herança de Propriedade, onde a classe mais especializada pode herdar todas as propriedades da classe mais geral. Associações entre *frames* determinam a sua estrutura hierárquica, ligando um *frame*-filho ao seu *frame*-pai. O *frame*-filho pode herdar valores de qualquer ancestral. A hierarquia de *frames* permite que dados sejam armazenados de maneira abstrata e com propriedades comuns compartilhadas, evitando assim a duplicação de dados, simplificando código e a manutenção.
- **Orientação a Objetos:** Reúne características presentes tanto das redes semânticas quanto dos *frames*. Na orientação a objetos a estratégia principal é representar o conhecimento como um conjunto de objetos e comportamentos. O conceito importante que difere dos demais é o de encapsulamento, que consiste em agrupar dados e métodos relacionados a um objeto, ocultando as informações de outros objetos. A flexibilidade é o ponto mais forte desses sistemas.
- **Orientação a Objetos Associada a Regras:** Combinação entre as técnicas de Regras de Produção e de Orientação Objeto, de forma a explorar ao máximo suas vantagens. A Orientação a Objeto pode fornecer uma forma rica, simples e natural para definir os objetos do domínio, suas relações e a forma de comportamento, as regras de produção oferecem um meio simples e natural de expressar o processo de raciocínio do sistema.

Para avaliar essas técnicas existem alguns critérios:

- Adequação lógica observa se o formalismo usado é capaz de expressar o conhecimento que se deseja representar;
- Conveniência notacional verifica as convenções da linguagem de representação. Se essas forem muito complicadas, a tarefa de codificação torna-se extremamente complexa.

(REZENDE, PUGLIESI e VAREJÃO, 2005, p. 30)

De acordo com Maia (2012, p. 112), “a representação do conhecimento é uma parte de vital importância para o programa, de tal forma que, se bem feito, não somente torna mais fácil a expansão do conhecimento residente, também, versatiliza a sua operação”.

Para Rezende, Pugliesi e Varejão (2005, p. 47), obter e representar todo o conhecimento de um determinado domínio é uma tarefa extremamente difícil. Sendo a aquisição do conhecimento uma tarefa manual que requer o auxílio de um usuário com *know-how* sobre o domínio (especialista), a falha em estruturar esse conhecimento ou a ausência de um tipo de conhecimento pode levar o SBC a gerar uma resposta não razoável.

Por conta deste problema, conhecido como a fragilidade dos SBC, este tipo de sistema é muito mais usado como assistente na resolução de problemas do que propriamente como os responsáveis pela tomada de decisão.

2.7 INTEGRAÇÃO ENTRE SISTEMAS

Para Junior (2008, p. 85), as organizações têm à sua disposição diversos tipos de SI para realizar os mais variados tipos de processos empresariais, visando aumentar o controle e a produtividade, como também fornecer suporte à decisão. Entretanto, cada um destes SI possui o seu banco de dados, em muitos casos não possuem uma integração, trabalhando de forma isolada.

Diante do cenário gerado pela ausência de integração, de acordo com Junior (2008, p. 85-86), os principais problemas gerados às organizações são:

- **Redundância de Dados:** Uma organização ao possuir diversos SI também possuirá diversas bases de dados, desta forma se torna comum que diversos dados existam em várias bases para atender

a inúmeros processos isolados, gerando uma grande redundância de tais dados.

- **Retrabalho:** Consiste na ação de incluir, modificar ou atualizar um determinado dado em mais de um SI. A principal consequência do retrabalho é à perda da produtividade por parte dos usuários em ter de digitar dados repetidos.
- **Falta de Integridade de Informações:** Devido aos dois problemas anteriores, este se torna o mais crítico, pois a informação fornecida pelo sistema pode ser conflitante ou não verdadeira. Este problema ocorre pela ação direta do fator humano, ou seja, se um usuário atualizar dados equivocadamente ou deixar de atualizá-los em uma ou mais de uma base, poderá comprometer a qualidade dos dados.

Para resolver os problemas apresentados acima, a integração sistêmica permite que sistemas separados se comuniquem diretamente um com o outro. Para Rezende (2005, p. 61), deve ser muito bem elaborada e estruturada a administração de dados, a existência de um módulo ou subsistema para realizar a gestão dos dados, pois torna mais eficiente à integração dos sistemas.

Rezende (2005, p. 30) define integração entre sistemas como sendo “as relações de interdependência entre os subsistemas que resultam na troca de informações entre eles”.

Na Tabela 6 são apresentados os benefícios advindos da integração de sistemas:

Tabela 6: Benefícios da Integração de Sistemas.

Benefícios	
Tangíveis	Intangíveis
- Redução de pessoal	- Aprimoramento dos processos
- Aumento de produtividade	- Padronização dos processos
- Aumento das receitas/lucros	- Satisfação dos clientes
- Entregas pontuais	- Flexibilidade e agilidade

Fonte: Adaptação de Junior (2008, p. 87).

Atualmente, com a grande presença da tecnologia nas organizações, recursos de tecnologia da informação, Rezende (2005, p. 31) considera que é muito improvável de encontrar subsistemas isolados e completamente independentes. Um

sistema integrado pelos vários subsistemas organizacionais que considerem as suas funções organizacionais internas.

De acordo com Turban, Wetherbe e Mclean (2007, p. 374), em um SAD comum, a ação de integrar outro componente, um Subsistema de Gestão do Conhecimento, como ilustrado na Figura 8, o torna mais avançado. Um SAD com este componente passa a denomina-se SAD inteligente, SAD/ES ou SAD baseado em conhecimento. Neste trabalho, iremos denominá-lo como Sistema Inteligente de Apoio à Decisão (SIAD).

O componente de GC fornece determinado conhecimento especialista para auxiliar na resolução de problemas não estruturados e semiestruturados complexos e permite também obter conhecimento para melhorar a operação de outros componentes do SAD.

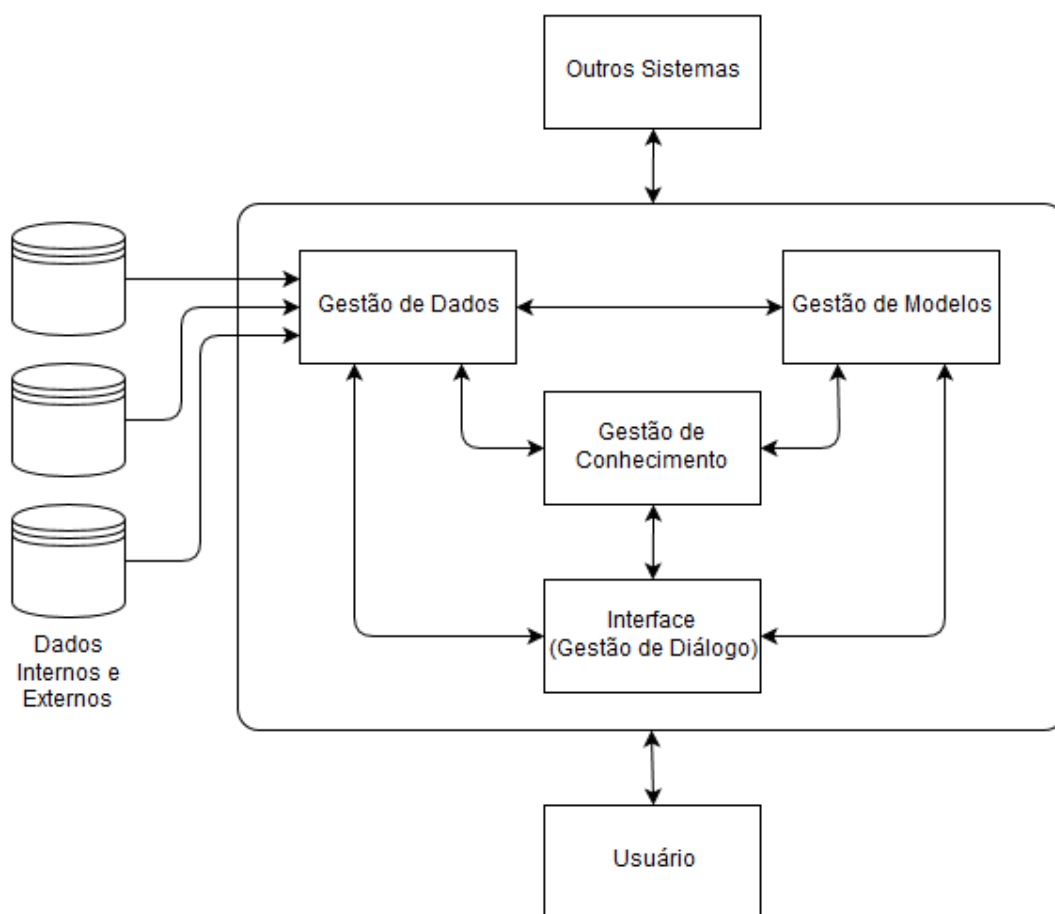


Figura 8: Estrutura de um SAD Inteligente.

Fonte: Adaptação de Turban, Wetherbe e Mclean (2007, p. 374).

A GC melhora a qualidade da tomada de decisão através da utilização do conhecimento. Para Laudon e Laudon (2011, p. 344), “as empresas operarão com menos eficácia e eficiência se o seu conhecimento particular não estiver disponível para a tomada de decisão e as operações correntes”.

No próximo capítulo será apresentado o AG, uma técnica da IA que pode ser aplicada aos SAD, são de fácil implementação, flexíveis e aplicáveis a uma grande variedade de problemas mantendo um ótimo desempenho.

3 ALGORITMOS GENÉTICOS

Neste capítulo, são apresentadas as referências teóricas relacionadas aos AG, os conceitos genéticos que servem de alicerces, a evolução histórica, estruturas e definições, seu funcionamento e demais características.

3.1 GENÉTICA

Nesta seção serão apresentados alguns conceitos e definições utilizados no campo da Genética, no qual se fazem necessários para um melhor entendimento das técnicas aplicadas aos AG. De acordo com Robinson (2015, p. 9), a genética pode ser definida como “o campo da ciência que examina como características são passadas de uma geração para outra”.

3.1.1 A Seleção Natural de Darwin

O conceito de AG vem da inspiração da teoria de Charles Darwin, descrita em sua obra intitulada “A Origem das Espécies”, publicada em 1859, no qual aborda um processo evolutivo das espécies, denominado de Seleção Natural.

Segundo Browne, Darwin descreveu o processo de seleção natural como:

Há uma guerra na natureza, um combate pela existência. Na luta para viver, se organismos piores ou mais fracos tendem a morrer primeiro, e as formas melhores, mais saudáveis ou mais bem adaptadas permanecem. Esses sobreviventes seriam aqueles que geralmente procriam. Se ações como essas se repetissem um número de vezes, os organismos tenderiam a se tornar cada vez mais bem adaptados às suas condições de existência.

(BROWNE, 2007, p. 52)

Com a reiteração de Linden (2008, p. 27), a teoria da evolução consiste da disputa entre os indivíduos de um ecossistema por comida e água. Os indivíduos de uma mesma espécie que não têm êxito em obter tais recursos, tendem a não ter

seus genes propagados, enquanto que os indivíduos que possuem maior êxito tem maior probabilidade de ter os seus genes presentes nas proles das novas gerações.

Para Linden (2008, p. 27), o acasalamento de indivíduos sobreviventes tem chance de gerar novos indivíduos mais bem adaptados às características de seu meio ambiente. Este processo implica que os descendentes são variações de seus pais, podendo apresentar ou não uma determinada característica.

De acordo com Robinson (2015, p. 328), Darwin quando publicou seu livro não tinha conhecimento do trabalho de Mendel, sobre as leis da herança genética, entretanto, mesmo sem o conhecimento de como funcionava as leis básicas da hereditariedade, fez referência com precisão de três princípios da genética:

- 1- A variação é aleatória e imprevisível;
- 2- A variação é herdável;
- 3- A variação muda de frequência com o tempo.

3.1.2 Leis da hereditariedade

Gregor Mendel, um monge e cientista com grande perícia em jardinagem, fundador da genética moderna. Em 1866, na cidade de Brunn, publicou um artigo baseado nos resultados obtidos de seus experimentos de jardinagem realizados com ervilhas. Através de simples observações, determinou com brilhante precisão como operava o mecanismo da herança genética. (ROBINSON, 2015).

Segundo Watson et al. (2015, p. 6-8), Mendel realizou vários cruzamentos entre progenitores diferentes entre si em apenas uma característica. Notou que toda a primeira geração de descendentes possuía característica de um dos progenitores. A característica que aparece nessa geração é chamada de dominante, enquanto que o traço ausente é denominado de recessivo.

Esses cruzamentos mostraram que a característica recessiva reaparecia na proporção de um quarto na segunda geração. Quando esses experimentos foram realizados até a terceira geração, todas as amostras recessivas da segunda geração geraram progênie pura. Os três quartos dominantes da segunda geração geraram um terço de progênie pura e dois terços de progênie mista, na proporção de 3:1 de dominantes para recessivos.

O termo fenótipo é utilizado quando se é referido à aparência ou a estrutura física de um indivíduo e o termo genótipo quando se refere à sua composição genética. O termo homocigoto refere-se a genes nos quais ambos os alelos, de ambos os progenitores, são idênticos. Enquanto o termo heterocigoto é para denominar os genes nos quais os alelos dos progenitores são diferentes. (WATSON, BAKER, *et al.*, 2015, p. 7).

A interpretação obtida por Mendel, que as diversas características são controladas por pares de fatores (chamados de genes), proveniente um de cada progenitor. Na Figura 9 apresenta o cruzamento entre um indivíduo da linhagem pura de ervilha lisa (RR) com outro da linhagem pura de ervilha rugosa (rr), a fertilização gera uma planta com a probabilidade de 100% para os alelos serem Rr na primeira geração.

Cross: $RR \times rr$		R	R
r	Rr	Rr	
r	Rr	Rr	

Figura 9: Cruzamento entre ervilhas lisas e rugosas.

A primeira geração será de híbridos, as sementes terão aparência lisa porque o alelo R é dominante sobre r . Então, Mendel realizou o cruzamento entre indivíduos desta geração e constatou que na segunda geração reaparecia o gene recessivo com a probabilidade de 25% e de 75% para dominante.

A Figura 10 apresenta a confirmação de que a proporção de dominantes para recessivos na segunda geração será sempre de 3:1.

Cross:
Rr × Rr

	R	r
R	RR	rR
r	Rr	rr

Figura 10: Cruzamento entre ervilhas híbridas lisas.

Segundo Watson et al. (2015, p. 7), “o reaparecimento da característica recessiva na segunda geração indica que os alelos recessivos não são modificados nem perdidos na primeira geração, mas que os genes dominantes e recessivos são transmitidos de forma independente” é denominado como primeira lei de Mendel ou princípio da Segregação Independente.

De acordo com Watson et al. (2015, p. 8), Mendel decidiu estender seus experimentos de cruzamento com ervilhas, buscando amostras que possuíam mais de uma característica de diferença. Mendel selecionou duas linhagens, uma das linhagens apresentava sementes lisas e amarelas e a outra com sementes rugosas e verdes. Na primeira geração concluiu que liso e amarelo são dominantes sobre rugoso e verde, pois novamente notou que toda a primeira geração de descendentes possuía característica de um dos progenitores.

As sementes da primeira geração foram cruzadas entre si, produzindo uma segunda geração, no qual os indivíduos foram analisados por seus fenótipos. Além dos fenótipos iniciais, surgiram dois novos tipos de recombinantes: rugoso e amarelo, e liso e verde.

Mendel achou que poderia interpretar os resultados pelo postulado dos genes. Na Figura 11 apresenta o cruzamento de duas linhagens puras de ervilhas, uma linhagem pura de sementes lisas e amarelas ($RRYY$) e outra linhagem pura de sementes rugosas e verdes ($rryy$), a fertilização tem 100% de probabilidade para gerar ervilhas lisas e amarelas ($RrYy$).

Cross:
RRYY × rryy

	RY	RY	RY	RY
ry	RrYy	RrYy	RrYy	RrYy
ry	RrYy	RrYy	RrYy	RrYy
ry	RrYy	RrYy	RrYy	RrYy
ry	RrYy	RrYy	RrYy	RrYy

Figura 11: Cruzamento entre ervilhas lisas e amarelas com rugosas e verdes.

A primeira geração será de híbridos, as sementes terão aparência lisa e amarela devido aos alelos *R* e *Y* serem dominantes. Um novo cruzamento com os indivíduos da primeira geração. Os fenótipos da progênie da segunda geração estão representados conforme a Figura 12.

Cross:
RrYy × RrYy

	RY	Ry	rY	ry
RY	RRYY	RRyY	rRYY	rRyY
Ry	RRYy	RRyy	rRYy	rRyy
rY	RrYY	RryY	rrYY	rryY
ry	RrYy	Rryy	rrYy	rryy

Figura 12: Cruzamento entre ervilhas híbridas lisas e amarelas.

Os quatro gametas possíveis são produzidos com frequências iguais e em cada par de genes era transmitido independentemente para o gameta durante a formação das células da linhagem germinativa, este é denominado como a segunda lei de Mendel ou o princípio da Distribuição Aleatória. (WATSON, BAKER, *et al.*, 2015, p. 8)

Segundo Henderson (2011, p. 11), nenhuma das duas leis é totalmente correta, existem exceções, entretanto, essas leis constituíram uma primeira tentativa de explicar a hereditariedade.

3.1.3 DNA e Cromossomos

DNA é a sigla para *Deoxyribonucleic Acid* ou Ácido Desoxirribonucleico em português. Toda a informação genética é armazenada em uma sequência linear de nucleotídeos no DNA. Cada molécula de DNA é uma dupla-hélice formada por duas fitas complementares de nucleotídeos que são mantidas unidas por ligações de hidrogênio entre os pares de bases. (ALBERTS, JOHNSON, *et al.*, 2010, p. 201).

A composição dos nucleotídeos é de açúcares, ligados a um único grupo fosfato e uma base. As bases são classificadas em dois tipos, purina e pirimidinas. As purinas são a adenina e a guanina, e as pirimidinas são a citosina e a timina. Os pares são sempre formados de adenina com timina e guanina com citosina. O DNA codifica a informação genética ao longo da fita, de forma que as bases são as letras desta mensagem. (WATSON, BAKER, *et al.*, 2015, p. 80).

De acordo com Alberts et al. (2010, p. 201), a duplicação da informação genética ocorre usando uma fita de DNA como molde para a formação da fita complementar. A informação genética contida no DNA de um organismo contém as instruções para todas as proteínas que o organismo irá sintetizar, compondo o genoma do organismo.

Para reiterar com Alberts, Linden (2008, p. 28) afirma que:

[...] todo indivíduo, seja ele animal, vegetal ou mesmo organismos inferiores como vírus e bactérias, é formado por uma ou mais células, e dentro de cada uma delas o organismo possui uma cópia completa de um ou mais cromossomos que descrevem o organismo, conjunto este denominado genoma.

Segundo Alberts et al. (2010, p. 202), um cromossomo é formado por uma molécula de DNA, extremamente longa, que contém inúmeros genes organizados linearmente. Watson et al. (2015, p. 199), acrescenta que a compactação de DNA em cromossomos serve para que o conteúdo se ajuste no interior da célula, proteger o DNA de lesões, ser transmitido de forma eficiente as células-filhas após o processo de divisão celular e confere organização geral para a molécula de DNA.

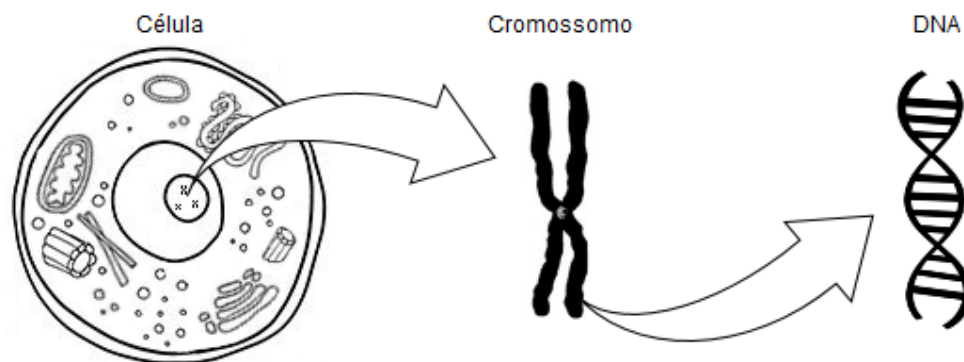


Figura 13: Célula, Cromossomo e DNA.

Um cromossomo consiste de genes, que são blocos de sequências de DNA, sendo que cada gene é uma região do DNA que tem uma posição específica no cromossomo, chamada *locus*. (LINDEN, 2008, p. 28). Na Figura 13 apresenta o DNA como parte do cromossomo localizado no núcleo da célula.

De acordo com Linden (2008, p. 28), os cromossomos são encontrados em pares, no núcleo da célula, sendo que o número de pares varia de espécie para espécie. Os genes presentes no cromossomo são blocos de sequências de DNA, sendo que cada gene é uma região do DNA com posição específica no cromossomo, chamada *locus*, que controla uma característica hereditária específica de cada organismo.

3.1.4 Recombinação Genética

Nos organismos que utilizam a reprodução sexuada que depende de um processo especializado de divisão nuclear chamado de meiose, que produz células haplóides. Estas células haplóides se diferenciam em células reprodutivas, cada progenitor fornece uma parte do seu material genético, chamado de gametas, como o óvulo e o espermatozóide. (ALBERTS, JOHNSON, *et al.*, 2010, p. 1090).

Segundo Alberts et al. (2010, p. 311), durante a meiose ocorre o processo de recombinação homóloga extensa que divide os cromossomos para as células germinativas. Existem dois processos que resultam na recombinação homóloga, a conversão gênica dos cromossomos e o *crossing-over* (entrecruzamentos), resultam desses eventos cromossomos híbridos com a informação genética dos homólogos materno e paterno.

Conversão gênica é o processo de reparo na divergência dos alelos na distribuição esperada durante a meiose. A Figura 14 apresenta quando ocorre do pareamento incorreto das hélices dos progenitores, o sistema de correção escolhe ao acaso qual a fita será corrigida, resultando na “conversão” de um alelo em outro. (ALBERTS, JOHNSON, *et al.*, 2010, p. 314)

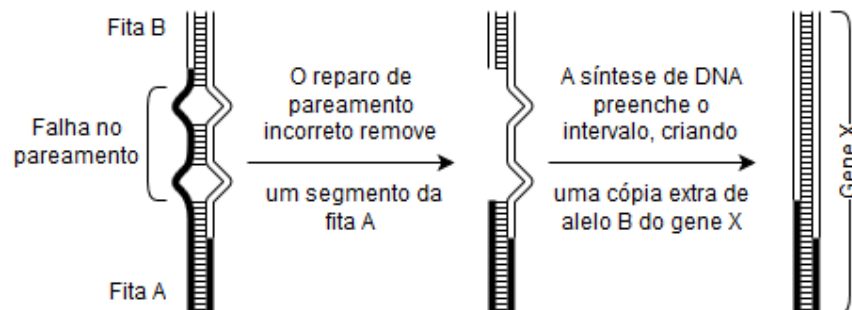


Figura 14: Exemplo de conversão gênica.

Fonte: Adaptado de Alberts et al. (2010, p. 315).

Alberts et al. (2010, p. 314-315) afirma que “vários processos celulares podem promover a conversão gênica”. Segundo Watson et al. (2015, p. 369), ocorre também quando um alelo de um gene é perdido e então ocorre a conversão gênica que substitui a lacuna por um alelo alternativo.

Crossing-over ou *crossover* é o processo de permuta de genes entre os cromossomos homólogos, ou seja, um pedaço de cada cromossomo é trocado com seu par para formar novas combinações, ocorrendo após a duplicação dos cromossomos durante a meiose. (WATSON, BAKER, *et al.*, 2015, p. 9).

Segundo Watson et al. (2015, p. 9), no início da meiose, por meio do processo de sinapse, os cromossomos homólogos formam pares, alinhando quatro cromátides (uma tétrade). Conforme a Figura 15, duas cromátides se entrelaçam formando um “x”, este ponto é denominado de quiasma, cada cromátide se quebra, ocorrendo à troca de genes ao se fundir com o segmento da outra.

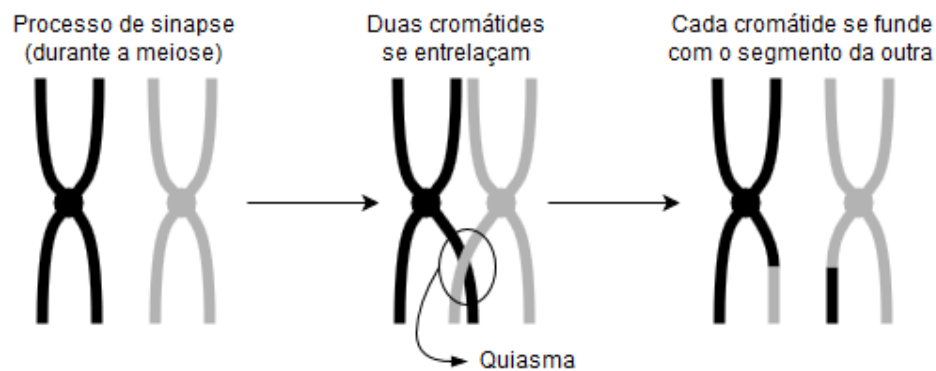


Figura 15: Processo de crossing-over.

Fonte: Adaptado de Watson et al. (2015, p. 11).

De acordo com Alberts et al. (2010, p. 1279), o processo de *crossing-over* contribui para a diversidade genética dos gametas. Com a distribuição aleatória dos homólogos dos progenitores, poderia produzir 2^n gametas geneticamente diferentes, onde n é o número de haploides de cromossomos. Porém, devido ao *crossing-over* o número real de gametas geneticamente diferentes é muito maior.

3.1.5 Mutação

A evolução das espécies está relacionada à capacidade de perpetuação dos indivíduos mais aptos ao ambiente que estão inseridos, os genes presentes são passados para seus descendentes através de gerações. Entretanto, podem ocorrer

erros nos genes durante a duplicação dos cromossomos. Segundo Alberts et al. (2010, p. 16-17), essas falhas são denominadas de mutações, que são alterações hereditárias na sequência de nucleotídeos de um cromossomo.

Segundo Watson et al. (2015, p. 13), o processo de mutação é raro de ocorrer, caso contrário, os descendentes de cada geração não seriam semelhantes aos seus progenitores. A ocorrência de uma pequena e finita taxa de mutação se torna vantajosa, pois fornece uma fonte constante de novas variantes, permitindo assim que plantas e animais se ajustem no caso de mudanças ambientais.

A mutação é um processo contínuo de seleção natural:

Por meio de intermináveis repetições desse ciclo de tentativas e erros – de mutação e seleção natural – os organismos evoluem: suas especificações genéticas mudam, proporcionando a eles novos caminhos para explorar o ambiente mais efetivamente, para sobreviver em competições com outros e para se reproduzir com sucesso. (ALBERTS, JOHNSON, *et al.*, 2010, p. 17).

Alberts et al. (2010, p. 16) afirma que em raras ocasiões a mutação pode representar mudanças para melhor, entretanto, na maioria dos casos, o erro pode acarretar em um sério dano ou a ausência de uma característica vital para sua sobrevivência. As mudanças que são favoráveis tendem ter mais chances de serem perpetuadas.

3.2 HISTÓRICO

Segundo Linden (2008, p. 34), a história dos AG se inicia na década de 40, quando cientistas começam a se inspirar na natureza para criar o ramo da IA. Na década de 50, iniciou a buscar modelos de sistemas genéricos que pudessem gerar soluções candidatas para difíceis problemas computacionais.

Em 1957, Box apresentou seu esquema de operações evolucionárias. No começo da década de 1960, Bledsoe e Bremmerman foram os pioneiros do trabalho operadores de recombinação (*crossover*) com genes em uma representação binária. Na metade dos anos 60, Rechenberg desenvolveu as estratégias evolucionárias, introduzindo a computação evolucionária às aplicações práticas. (LINDEN, 2008).

No final da década de 60, John Holland, designado como o pai dos AG, inseriu os AG, embora concentrado eminentemente na codificação discreta. Holland

estudou a evolução das espécies e propôs um modelo heurístico computacional. Em 1975, publicou o livro “*Adaptation in Natural and Artificial Systems*”, apresentando os AG como uma metáfora para os processos evolutivos, no qual permitia estudar a adaptação e a evolução do mundo real. (LINDEN, 2008, p. 34-35).

Nos anos 80, além do progresso científico, iniciou-se o desenvolvimento comercial com os AG. Com o progresso dos algoritmos evolucionários, surgiram as primeiras conferências dedicadas exclusivamente a este tópico. (LINDEN, 2008).

3.3 DEFINIÇÃO E CARACTERÍSTICAS

Esta seção será iniciada com a terminologia dos Algoritmos Genéticos (AG). Como os AG são fortemente inspirados na teoria da evolução das espécies e na genética, os termos utilizados são analogias aos termos empregados na biologia. (LINDEN, 2008, p. 46). Na Tabela 7 são apresentados os termos utilizados nos AG:

Tabela 7: Terminologia utilizada em AG.

Termos	Descrição
Cromossomos ou Indivíduos ¹³	É o conjunto de genes que totalizam a informação total de um indivíduo. Pode ser representado de forma binária ou <i>string</i> (de bits). Cada uma possível solução do problema é representada por um cromossomo.
Gene ou variável	É a unidade básica de informação. Representa uma característica que descreve parte do problema a ser solucionado.
Alelo	O valor de um gene.
Lócus	A posição do gene no cromossomo.
Genótipo	Representação interna usada pelo AG para armazenar os parâmetros a serem otimizados.
Fenótipo	Consiste nos valores reais dos parâmetros e sua inserção no problema sendo resolvido.
Geração	Corresponde ao número de iterações realizadas pelo AG.
Operadores Genéticos	Consistem em aproximações computacionais de fenômenos vistos na natureza, como a reprodução sexuada, a mutação genética e quaisquer outros.
Avaliação	É a quantificação da qualidade de um cromossomo como solução do problema em questão.
<i>Fitness</i>	É o valor atribuído a um cromossomo ao se avaliar a melhor adequação do seu fenótipo ao meio ambiente.

¹³ Há autores que se referem a indivíduos ou estados (RUSSELL e NORVIG, 2013), outros fazem referência ao termo cromossomos (LACERDA e CARVALHO, 1999). Para Linden (2008, p. 46), os dois termos, cromossomos e indivíduos, são intercambiáveis.

Fonte: Adaptado de Linden (2008, p. 40-41,47).

Linden (2008, p. 43) define os AG como “uma técnica de busca baseada numa metáfora do processo biológico de evolução natural”. Os AG se utilizam de conceitos do campo da genética para que de forma simulada ocorra à evolução da população de possíveis soluções.

De forma resumida, o funcionamento dos AG ocorre da seguinte maneira: populações de indivíduos são criadas e submetidas a operadores genéticos. Cada indivíduo é avaliado e no caso de ser apto, participará de um processo de evolução, que eventualmente deverá gerar um indivíduo que caracterizará uma boa solução para o problema em questão. (LINDEN, 2008, p. 43-44).

Para Linden (2008, p. 44) no caso dos AG, o ambiente que os indivíduos vivem é permanentemente o mesmo, somente as gerações de indivíduos vão sendo modificadas devido à ação dos operadores genéticos. O processo consiste em fazer uma competição entre os indivíduos pela sobrevivência do gene mais apto. Com a sobrevivência do mais apto, os melhores indivíduos prevalecerão.

AG são técnicas probabilísticas, e não determinísticas. Assim sendo, com uma mesma população inicial e mesmo conjunto de parâmetros é possível de obter soluções diferentes a cada vez que é executado o algoritmo. (LINDEN, 2008, p. 48).

Os AG são em geral programas simples que trabalham com uma grande população de pontos, sendo uma heurística de busca no espaço de solução e nunca uma busca significativa das soluções possíveis. A sua busca utiliza as informações pertinentes ao problema e direcionam sua busca através do mecanismo de seleção.

Segundo Linden (2008, p. 49), os AG trabalham com a forma codificada dos parâmetros na representação cromossomial a serem aprimorados e não com os parâmetros diretamente.

Toda a informação relativa ao problema está contida na função de avaliação, que realiza a codificação e decodificação dos parâmetros. Desta forma, um AG pode ser utilizado para atender a uma infinidade de problemas apenas remodelando a função de avaliação. (LINDEN, 2008, p. 49).

3.4 ESTRUTURA BÁSICA

Para se implementar um AG, o modelo computacional deve observar o conceito de simulação da evolução das espécies através da seleção, mutação e reprodução, processos estes que dependem do “desempenho” dos indivíduos desta espécie dentro do “ambiente”. (LINDEN, 2008, p. 40).

A estrutura básica de funcionamento dos AG é apresentada na Figura 16:

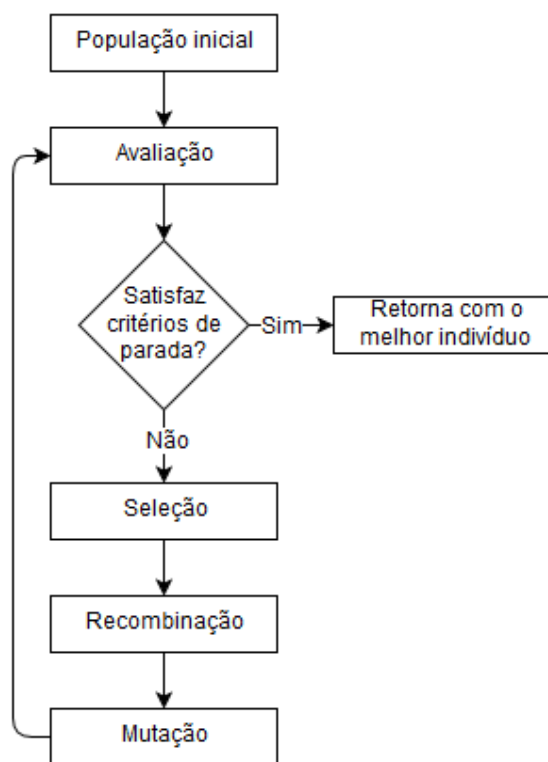


Figura 16: Estrutura de funcionamento de um AG.

Fonte: Adaptado de Linden (2008, p. 57).

De acordo com Linden (2008, p. 57), os AG são altamente genéricos, pois vários componentes são invariáveis de um problema para outro. Então se um AG for implementado em uma linguagem orientada a objeto, permitirá o reaproveitamento do código para solução de vários problemas diferentes.

As próximas subseções serão para explicar cada uma destas etapas da estrutura básica do AG.

3.4.1 População Inicial

De acordo com Lacerda e Carvalho (1999, p. 90-91), “um cromossomo é uma estrutura de dados, geralmente um vetor ou cadeia de bits, que representa uma possível solução do problema a ser otimizado”.

A população de um AG corresponde as possíveis soluções do problema e a inicialização da população consiste de escolher n indivíduos. A forma mais utilizada de inicializar a população é de forma aleatória, entretanto, também é possível criar indivíduos com características pré-selecionadas. (LINDEN, 2008).

Ao iniciar uma população aleatória, para Linden (2008, p. 64), a escolha dos indivíduos será baseada na lei das probabilidades, haverá uma distribuição que cobre praticamente todo o espaço de soluções, entretanto, não pode ser garantido, pois a população tem um tamanho finito diferente do número total de arranjos.

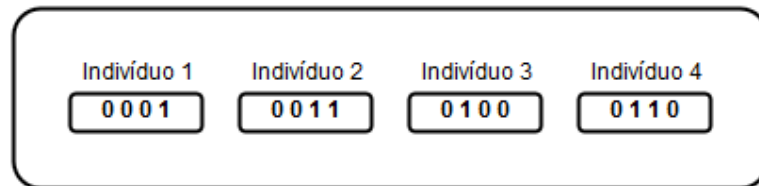


Figura 17: População de quatro indivíduos.

Fonte: Adaptação de Linden (2008).

De acordo com Russel e Norvig (2013, p. 115), o indivíduo é representado por uma cadeia de caracteres, a forma mais frequente de representação é por meio da cadeia binária, formada por valores “0” e “1”.

Na Figura 17, um cromossomo que possua um alelo com o valor “1” indica que possui tal característica enquanto que o valor “0” não a possui. A representação binária é uma boa opção, pois para ela os operadores genéticos são extremamente simples e o computador lida com ela de forma natural. (LINDEN, 2008).

3.4.2 Avaliação

A função de avaliação¹⁴ tem como objetivo determinar a qualidade de um indivíduo como solução do problema. De acordo com Linden (2008, p. 65) e Russel e Norvig (2013, p. 115), a função concede uma nota baseada nas características do indivíduo, de forma que ao retornar valores mais altos para indivíduos melhores, permitindo diferenciá-lo como uma boa ou má solução.

Em muitos casos, Linden (2008, p. 65) afirma que a função de avaliação é a única ligação do AG com o problema real. Um mesmo AG pode ser utilizado para a resolução de vários problemas, a única estrutura do algoritmo que deve ser ajustada ao problema será a função de avaliação.

Indivíduos	Avaliação
0 0 0 1	1
0 0 1 1	9
0 1 0 0	16
0 1 1 0	36

Figura 18: Avaliação dos indivíduos.

Fonte: Adaptação de Linden (2008).

Para Linden (2008, p. 66-67), a importância da função de avaliação como o componente responsável por atribuir valores que indicam a qualidade do indivíduo deve ser elaborada com muito cuidado e baseada em todo o conhecimento que envolve o problema, tanto as restrições quanto seus objetivos de qualidade. Na Figura 18 apresenta um exemplo de avaliação dos indivíduos.

Os valores atribuídos pela função de avaliação são calculados com base na aproximação entre objetivos e restrições relacionados com o problema. No caso da função não ser bem elaborada, é possível atribuir um valor de pouca precisão ao indivíduo, em consequência poderá ocasionar aumento no tempo de busca por um candidato já descartado. (LINDEN, 2008).

¹⁴ A função pode ser denominada como: função de avaliação ou função de custo (LINDEN, 2008), função de *fitness* (RUSSELL e NORVIG, 2013) e função objetivo (LACERDA e CARVALHO, 1999).

3.4.3 Seleção

A etapa de seleção deve simular o mecanismo de seleção natural, no qual os indivíduos de uma população formam pares, de forma que os pais mais capazes geram mais proles, enquanto pais menos aptos podem ou não gerar progênie.

Segundo Russel e Norvig (2013, p. 115), “a probabilidade de um indivíduo de ser escolhido para reprodução é diretamente proporcional à sua pontuação de *fitness*”. Para Linden (2008, p. 67), deve-se privilegiar os indivíduos com *fitness* mais elevado, entretanto, não desprezar completamente os que são menos aptos.

De acordo com Linden (2008, p. 67-68), se a reprodução ocorrer sempre entre os melhores indivíduos, a população tenderá a ser composta por indivíduos mais semelhantes e faltará diversidade, este efeito é denominado de convergência genética. Para eliminar esse efeito ou minimizá-lo, os indivíduos menos aptos devem ser selecionados de forma justa participando do processo de seleção.

Existem vários métodos para a seleção dos indivíduos:

- **Método da Roleta:** É um método de seleção bem simples e o mais utilizado. Neste método os indivíduos são selecionados através de uma roleta, um conceito abstrato, semelhante à roleta utilizada em jogos de azar. Na Figura 19 apresenta os indivíduos da população representados na roleta por uma fatia proporcional ao seu *fitness*, isto é, maior *fitness* ocupa uma fatia maior, enquanto menor *fitness* fatia menor. (CARVALHO, BRAGA e LUDERMIR, 2005, p. 233).

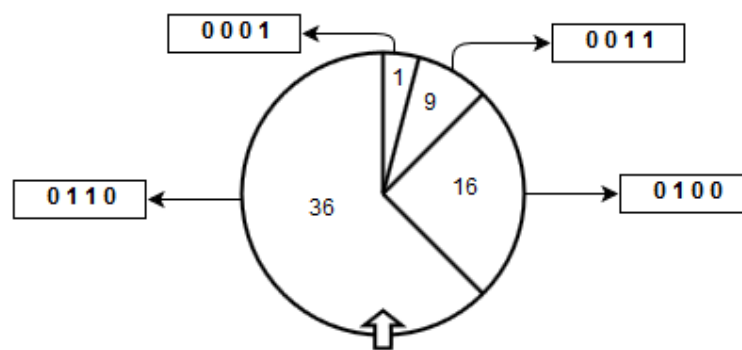


Figura 19: Método da Roleta.

Fonte: Adaptação de Linden (2008).

- Método de Seleção Local:** Neste método, cada indivíduo existe em uma vizinhança e suas fronteiras determinam os indivíduos com que podem interagir. A vizinhança pode possuir uma estrutura com qualquer número de dimensões e uma distância limite para indicar que um indivíduo ainda é considerado pertencente. As vizinhanças podem ter interseções. Na Figura 20 apresenta um exemplo do método de Seleção Local. (LINDEN, 2008, p. 177).

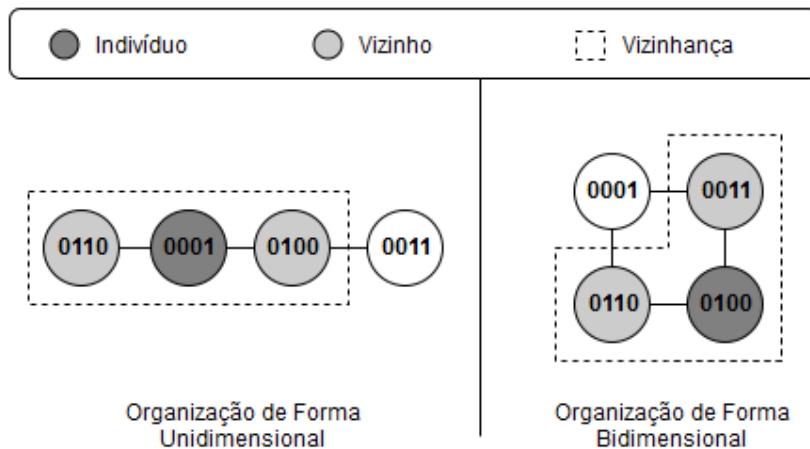


Figura 20: Seleção Local.

Fonte: Adaptação de Linden (2008, p. 177).

- Método do Torneio:** É o método onde n indivíduos da população são escolhidos aleatoriamente. O cromossomo com maior *fitness* é selecionado. Geralmente, o valor utilizado de n é três. O processo de seleção é repetido até que se alcance o número de indivíduos da população. Na Figura 21 apresenta um exemplo do Método de Torneio. (CARVALHO, BRAGA e LUDERMIR, 2005, p. 234).

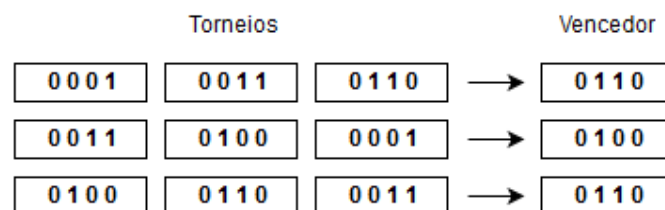


Figura 21: Método do Torneio.

- **Método da Amostragem Universal Estocástica:** É uma variação do método da roleta, em que são adicionadas n agulhas em vez de uma única agulha. As n agulhas são igualmente espaçadas, onde n é o número de indivíduos a serem selecionados. Desta forma a roleta é girada apenas uma única vez, exibindo menos variância, conforme a Figura 22. (CARVALHO, BRAGA e LUDERMIR, 2005, p. 234-235).

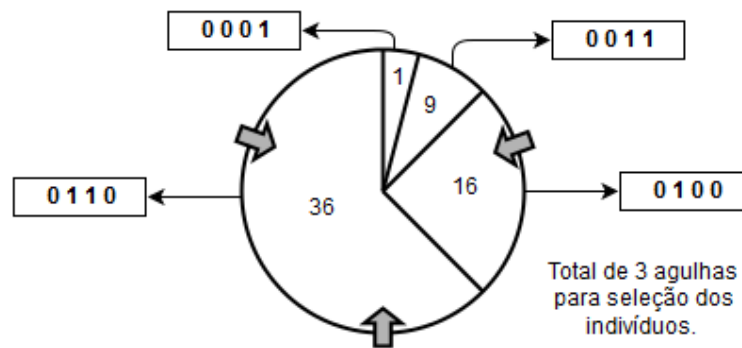


Figura 22: Método da Amostragem Universal Estocástico.

- **Método da Seleção Truncada:** Neste método apenas os melhores $x\%$ da população poderão ser escolhidos como pais da próxima geração. Os indivíduos são ordenados de forma decrescente com base no valor de sua avaliação e os que estiverem dentro do percentual poderão participar da seleção. Este método causa a convergência genética no caso de x ser um número pequeno, devido ao fato de causar maior pressão seletiva do que os outros métodos. O lado positivo é eliminar com mais rapidez os indivíduos menos aptos, fazendo com que o AG se concentre nas melhores soluções. Na Figura 23 apresenta o exemplo da utilização deste método. (LINDEN, 2008, p. 181).

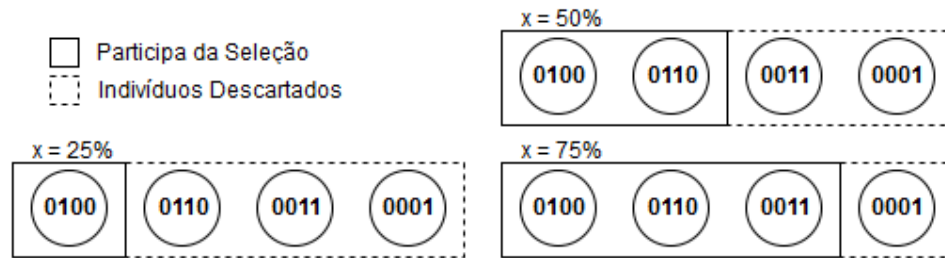


Figura 23: Método da Seleção Truncada.

3.4.4 Recombinação

Com os progenitores escolhidos na etapa de seleção, nesta fase ocorre o processo de recombinação dos genes. Para Linden (2008, p. 74) e Carvalho, Braga e Ludermir (2005, p. 96), para cada par de progenitores a ser cruzado é escolhido ao acaso um ponto de *crossover*, ou seja, um ponto de corte dentre as posições na cadeia de genes.

Para Linden (2008), “um ponto de corte constitui uma posição entre dois genes”. Cada indivíduo de n genes contém $n - 1$ pontos de corte. O ponto de corte é o ponto de separação do material genético de um dos progenitores para que seja recombinado com o material do outro progenitor, gerando duas proles.

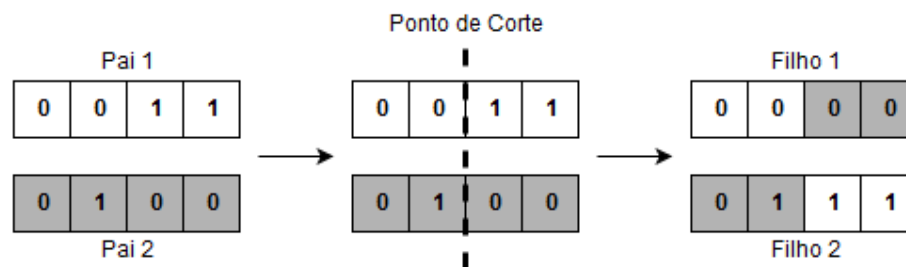


Figura 24: *Crossover* de um ponto.

Fonte: Adaptação de Linden (2008, p. 77).

Conforme ilustrado na Figura 24, o filho 1 é formado pela concatenação da parte esquerda do pai 1 com a parte direita do pai 2 e o filho 2 pela concatenação

das duas partes restantes. Afirma Linden (2008, p. 75), que este processo é similar com o processo que ocorre na natureza durante a reprodução sexuada.

De acordo com Linden (2008, p. 74), o processo descrito acima é um tipo de *crossover*, extremamente simples, denominado de ***crossover de um ponto***, pelo fato de ocorrer apenas um ponto de corte.

Existem outros tipos de operadores de recombinação, mais complexos e eficientes que serão descritos a seguir:

- **Crossover Multipontos:** O processo é similar ao *crossover* de um ponto, com o acréscimo de x pontos de cortes. O objetivo é a maior permuta dos $x + 1$ fragmentos do material genético entre os progenitores. (CARVALHO, BRAGA e LUDERMIR, 2005, p. 237). Na Figura 25 apresenta o caso de *crossover* para dois pontos, o filho 1 será formado pela parte do pai 1 fora dos dois pontos de corte e pela parte do pai 2 entre os pontos de corte e o filho 2 pela junção das partes restantes. (LINDEN, 2008, p. 117).

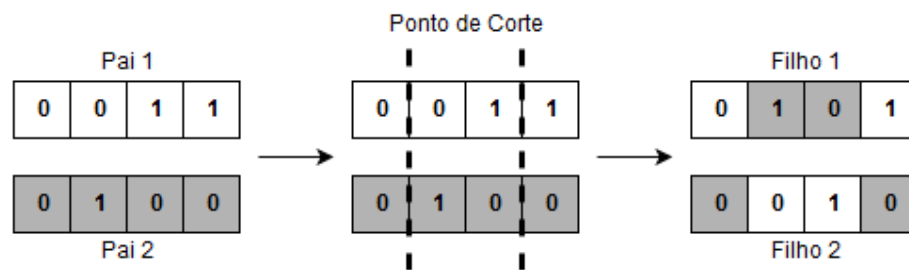


Figura 25: Crossover Multipontos.

Fonte: Adaptação de Linden (2008, p. 117).

- **Crossover Uniforme:** Este operador não se baseia em pontos de corte, seu funcionamento é realizado por meio de uma máscara, na qual indica quais genes serão destinados dos progenitores para suas proles. A máscara é montada por sorteio para cada gene. Se o valor sorteado for igual a um, o filho 1 recebe o respectivo gene do pai 1 e o filho 2 recebe do pai 2, no caso de ser sorteado o valor zero as atribuições são invertidas, o filho 1 recebe o gene do pai 2 e o filho 2 recebe do pai 1. Na Figura 26 apresenta exemplo do uso do Crossover Uniforme. (LINDEN, 2008, p. 119-120).



Figura 26: *Crossover Uniforme*.

Fonte: Adaptação de Linden (2008, p. 120).

- **Crossover Baseado em Maioria:** Não é muito usado devido a sua tendência de causar rapidamente a convergência genética. Este *crossover* consiste em sortear n pais e fazer com que o alelo do filho seja igual ao valor da maioria dos pais selecionados. Na Figura 27 apresenta exemplo do *Crossover* baseado em Maioria.



Figura 27: *Crossover Baseado em Maioria*.

Fonte: Adaptação de Linden (2008, p. 121-122).

3.4.5 Mutação

Após a formação das proles na etapa de recombinação, entra em ação o operador de mutação. Este operador atua na nova população em uma probabilidade muito baixa. Segundo Linden (2008, p. 77), a taxa de mutação deve ser baixa e estar na ordem de 0,5%, entretanto, Lacerda e Carvalho (1999, p. 97), afirmam que a taxa a ser usada se encontra entre 0,1% e 5%, o suficiente para assegurar a diversidade.

Segundo Lacerda e Carvalho (1999), o operador de mutação melhora a diversidade dos genes da população, no entanto, se não utilizada devidamente, pode destruir uma característica presente na população. Para Linden (2008), não há um valor ideal para ser fixado sobre a probabilidade que decide se o operador será ou não aplicado, mas recomenda que seja baixa conforme no mundo real.

A cada gene é verificada a probabilidade de ocorrer mutação, no caso ser positivo, o operador atua sobre o gene alterando o valor. Segundo Linden (2008), há um sorteio entre os valores 0 e 1, ou seja, 50% de chance do gene ser alterado para um outro valor, enquanto que para Lacerda e Carvalho (1999, p. 97) o valor do gene é invertido, ou seja, 100% de ser alterado.

O operador de mutação é uma heurística exploratória que injeta novos genes na população e permitindo que o AG busque soluções fora dos limites definidos pela população inicial. Então, a ocorrência de mutação aumenta a chance de se alcançar qualquer ponto do espaço de busca. (LINDEN, 2008).

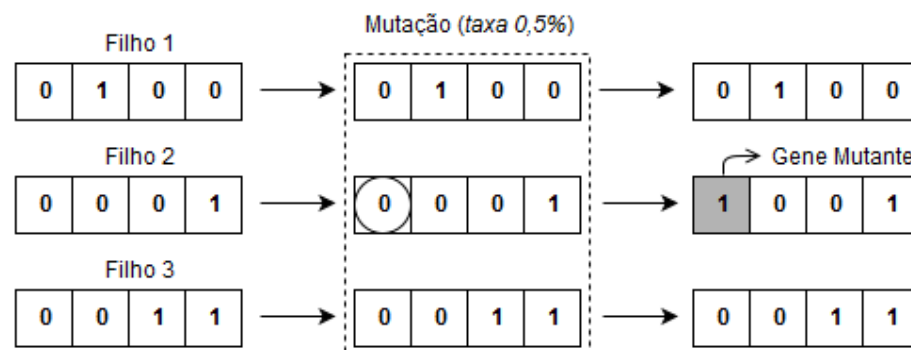


Figura 28: Mutação.

Como verificamos na etapa de *crossover*, com a combinação dos pais: 0001, 0011, 0110 e 0100, foram obtidas várias proles que apresentavam as mesmas características encontradas nos progenitores. Entretanto, somente com o operador de mutação, foi possível alcançar uma característica nova que não estava presente na população inicial (ver Figura 28).

3.4.6 Parâmetros de Controle

Os parâmetros de controle podem ser considerados como os elementos vitais para a execução e a eficiência do algoritmo genético. Os parâmetros descritos nesta seção são considerados *default* entre os AG. De acordo com Linden (2013, p. 61 e 64) e (2008), os parâmetros são os seguintes:

- **Tamanho da População:** Número de indivíduos que pertencem à população através das gerações, este parâmetro normalmente é constante ao longo da execução do AG.
- **Taxa de Recombinação:** Indica a probabilidade de recombinação entre os indivíduos selecionados na população. Esta taxa indica a pressão do ambiente sobre os indivíduos menos aptos na chance de passar seus genes às próximas gerações.
- **Taxa de Mutação:** Indica a probabilidade de ocorrer mutação nos genes dos indivíduos.
- **Número de Gerações:** Indica o número de gerações em que o AG irá utilizar como critério de parada.

3.4.7 Critério de Parada

O AG realiza um número indefinido de iterações, em que cada repetição do algoritmo corresponde a uma nova geração de indivíduos. Caso o algoritmo não alcance um critério de parada ou se não existir um para tal, o algoritmo continuará a ser executado indefinidamente.

Para encontrar o melhor indivíduo da população, ou seja, a solução do problema, o AG deve possuir pelo menos um critério de parada, que seja possível de ser alcançado, estabelecendo assim a saída da iteração do algoritmo.

De acordo com Linden (2008, p. 82) e Lacerda e Carvalho (1999, p. 125), a seguir são listados alguns exemplos de critérios de parada:

1. Caso o AG alcance um número x de gerações decorridas, o melhor indivíduo (melhor *fitness*), retorna como solução;

2. Se for obtido um indivíduo com valor de avaliação melhor ou igual a um valor x pré-determinado, este indivíduo retorna como solução;
3. Se o AG está sendo executado por um tempo decorrido superior a x , retorna o melhor indivíduo.
4. Quando ocorrer um número x de gerações sem melhora no *fitness* da população corrente, retorna o melhor indivíduo.
5. Se a característica de um gene está presente acima dos 90% da população, retorna o melhor indivíduo.

Os critérios de parada podem ser simples ou representar um conjunto de regras mais elaboradas. Os critérios devem ser analisados cuidadosamente, pois sempre existirá a chance de ocorrer convergência genética na população e a solução não será a melhor para o problema.

3.5 TEOREMA DOS ESQUEMAS

O teorema dos esquemas foi enunciado por John Holland, em 1975, no qual procura fundamentar o comportamento dos AG. Holland verificou que os AG manipulam nos cromossomos determinados segmentos de genes, cadeia de bits, denominados de esquemas. (LACERDA e CARVALHO, 1999, p. 104).

Segundo Mitchell (1996) citado por Linden (2008, p. 107), pela análise de Holland, um AG calcula a avaliação de n indivíduos da população atual, mas que ele calcula a avaliação de um número bem maior de esquemas instanciados por cada indivíduo da população.

Holland mostrou também que esquemas com avaliações superior tendem a ocorrer com mais frequências nas próximas gerações e esquemas com avaliações abaixo da média tendem a desaparecer. (LINDEN, 2008).

Um esquema é formado por 0, 1 e *. A ocorrência do * em uma posição no esquema, indica que esta posição pode ser ocupada por 0 ou 1. O número total de esquemas pode ser obtido por 2^g , onde g é o número de genes. (LACERDA e CARVALHO, 1999)

Tabela 8: Esquemas.

Esquemas Cromossomos	1 * * * * (e1)	** 1 0 * (e2)	* 0 * 0 1 (e3)
1 1 0 0 1 (c1)	X	-	-
1 1 0 1 1 (c2)	X	-	-
1 0 1 0 1 (c3)	X	X	X

Fonte: Adaptação de Lacerda e Carvalho (1999, p. 104).

Na Tabela 8, verificamos que os esquemas $e1$, $e2$ e $e3$ estão inseridos no cromossomo $c3$, que ao todo pode ter $2^5 = 32$ esquemas. E também que o esquema $e1$ está inserido nos cromossomos $c1$, $c2$ e $c3$. (LACERDA e CARVALHO, 1999).

De acordo com Linden (2008), o teorema diz que, sendo n o número de indivíduos pertencentes a um esquema s , com média de avaliação igual a r e sendo x a média das avaliações de toda a população, então o número de ocorrências s na próxima geração é aproximadamente igual a $n * r/x$.

Segundo Lacerda e Carvalho (1999, p. 105), pela observação da equação é possível concluir que se $r > x$, ou seja, se a média de avaliação estiver acima da média das avaliações de toda a população, o esquema s pode ser destruído ao passar para a próxima geração.

Os operadores de *crossover* e mutação podem destruir os esquemas. No *crossover*, caso o corte ocorra no meio do esquema irá destruí-lo para sempre. E na mutação, caso ocorra em uma posição no qual o gene é diferente de *, também irá destruir o esquema. (LINDEN, 2008).

De acordo com Linden (2008, p. 109), o teorema dos esquemas expressa que os AG “tendem a preservar com o decorrer do tempo aqueles esquemas com maior avaliação média e com menores ordem e tamanho, combinando-os como blocos de armar de forma a buscar a melhor solução”.

3.6 ELITISMO

O Elitismo é uma estratégia para que os melhores n indivíduos de cada geração não devem “morrer” junto com a sua geração, mas o de continuar existindo na próxima geração visando preservar a qualidade de seu genoma. (LINDEN, 2008).

Esta estratégia é muito comum nos AG tradicionais, em transferir o melhor indivíduo de uma geração para outra, desta forma a melhor solução encontrada até o momento não será descartada devido ao corte do *crossover* ou à ocorrência de mutação. (LACERDA e CARVALHO, 1999, p. 99).

Segundo Linden (2008, p. 141), o Elitismo garante que o desempenho do AG seja crescente conforme as gerações vão passando (abaixo, ver Gráfico 1). A maioria dos esquemas de avaliação de desempenho dos AG mede apenas a adequação da melhor solução dentre todos os indivíduos.

Para Linden (2008) a manutenção do melhor indivíduo da geração t na população da próxima geração $t + 1$, garante que o melhor indivíduo de $t + 1$ é pelo menos igual ao melhor indivíduo da geração anterior. Garantindo assim que o gráfico da avaliação do melhor indivíduo como função do número de gerações decorridas seja uma função crescente.

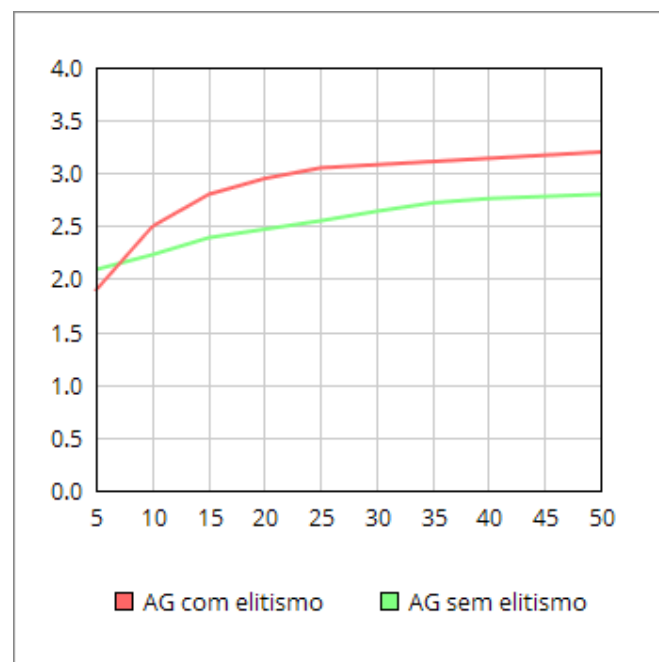


Gráfico 1: Desempenho do AG.

Fonte: Adequação de Lacerda e Carvalho (1999, p. 100).

No Gráfico 1, claramente é mostrado que o AG com elitismo encontra uma solução mais rápida que o AG sem elitismo. Isto ocorre nos AG sem elitismo devido ao fato dos máximos locais serem encontrados em algumas gerações,

tornando a média dos melhores indivíduos menor, enquanto que nos AG com elitismo em todas as gerações haverá pelo menos um máximo local elevando a média dos melhores.

No próximo capítulo será apresentada a LF, outra técnica da IA que pode ser aplicada aos SAD. Esta técnica facilita a especificação de regras em linguagem próxima à natural, baseando-se em palavras do que em números, de forma a tratar imprecisões mais próximas do raciocínio humano para simplificar a resolução de problemas.

4 LÓGICA FUZZY

Neste capítulo são apresentadas as referências teóricas relacionadas com a LF, os conceitos e definições, seu funcionamento e demais características.

4.1 CONCEITO

Os seres humanos tendem a categorizar tudo, usando regras para tomar decisões que podem possuir muitas nuances¹⁵ de significado (LAUDON e LAUDON, 2011, p. 338). Por exemplo, ao analisar uma determinada distância, é possível que um indivíduo a categorize como “perto”, um segundo como “longe” e um terceiro indivíduo considere “muito longe”.

Uma mesma medida de distância foi categorizada de várias maneiras, isto ocorre devido ao fato que essa categorização ocorre de forma subjetiva e imprecisa, uma característica do raciocínio humano.

A LF, também conhecida como Nebulosa ou Difusa, é um ramo da matemática que lida com incertezas baseada no raciocínio humano, refletindo a forma como pensamos, tentando modelar uma resposta mais adequada à realidade. (RAINER e CEGIELSKI, 2012, p. 329).

Segundo Perez e León (2007, p. 12), a LF é uma ferramenta baseada em regras com valores aproximados ou subjetivos para representar tais incertezas, em vez de suprimir as com imposições arbitrárias da lógica bivalente.

Em um contexto geral, a LF é mais poderosa e dinâmica do que a lógica convencional ou clássica, pois admite que uma resposta possa ser mais apropriada a um problema, como por exemplo, um “talvez”, do que escolher entre “verdadeiro” ou “falso”. Desta forma, a LF permite que a decisão computacional se aproxime da decisão humana.

¹⁵ Do Francês *nuece*. Significa a diferença tênue entre coisas do mesmo gênero.

4.2 HISTÓRICO

Segundo Perez e León (2007, p. 11), a precisão exibida pela matemática deve seu sucesso em grande parte aos esforços de Aristóteles (384-322 A.C) e os filósofos gregos que o precederam. Esforços que hoje são conhecidos como as Leis do pensamento.

Uma das leis, o Princípio do Terceiro Excluído, formulado por Aristóteles, afirma que qualquer proposição deve ser verdadeira ou falsa. Parmênides (530-460 A.C.) propôs a primeira versão desta lei. E no mesmo período, contrário a esta lei, Heráclito (540-470 A.C.) afirmou que simultaneamente uma proposição poderia ser verdadeira ou não. (PEREZ e LEÓN, 2007).

Platão propôs a existência de uma terceira região além do verdadeiro ou falso, onde as extremidades são unidas, se tornando bases do que seria conhecido como LF. Perez e León (2007, p. 12) afirmam que outros filósofos modernos como Hegel, Marx e Engels também revelaram seus pontos de vista, mas foi Lukasicwicz que propôs uma alternativa sistemática à lógica bivalente de Aristóteles.

Por volta de 1900, Lukasicwicz descreveu uma lógica de três valores, que além do verdadeiro e falso, havia a presença do valor intermediário com significado de “possível”, ou seja, o conjunto dos valores 0, 0.5 e 1. Com o aperfeiçoamento do estudo, Lukasicwicz criou a notação polonesa e os sistemas lógicos polivalentes. Mais tarde, propôs a lógica de valores infinitos entre 0 e 1. (PEREZ e LEÓN, 2007).

Em 1965, Lotfi Zadeh publicou o trabalho *Fuzzy Sets* no qual apresentou a teoria dos conjuntos difusos. Conforme Perez e León (2007), esta teoria propõe a extensão da lógica clássica através de uma função no qual as extremidades seriam representadas pelos valores verdadeiros e falsos, operando ao longo do intervalo dos números reais 0 e 1. (WEBER e KLEIN, 2003, p. 23).

No ano de 1973, Zadeh citado por Weber e Klein (2003), apresentou o princípio da incompatibilidade:

“À medida que a complexidade de um sistema aumenta, nossa habilidade para fazer afirmações precisas e que sejam significativas acerca deste sistema diminui até que um limiar é atingido além do qual precisão e relevância tornam-se quase que características exclusivas”.

No ano seguinte, o professor Ebrahim Mamdani, do *Queen Mary College*, Universidade de Londres, após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos distintos de controladores, somente o conseguiu fazê-lo através de aplicações do raciocínio *fuzzy*. (WEBER e KLEIN, 2003, p. 23).

O sucesso de Mamdani serviu de alavanca para muitas aplicações, por exemplo, os controles *fuzzy* de plantas industriais, refinarias, processos biológicos e químicos, tratamento de água, sistema de operação automática de trens, trocador de calor e máquina diesel. A primeira área a ser beneficiada com o surgimento LF foi a de sistemas de controle. (WEBER e KLEIN, 2003).

Zadeh é considerado o grande colaborador¹⁶ da LF, no qual combinou os conceitos da lógica clássica e os conjuntos de Lukasicwicz. Segundo Weber e Klein (2003, p. 24), na atualidade a LF é uma das tecnologias mais poderosas aplicadas em sistemas de controle e de apoio à decisão devido à sua capacidade de tratar incertezas.

4.3 CONJUNTOS FUZZY

O conceito da teoria de conjuntos *fuzzy* foi introduzido por Zadeh (1965) para possibilitar a modelagem de sistemas que não eram possíveis de tratar conjuntos de informações subjetivas e imprecisas com os métodos e as abordagens clássicas. (PEREZ e LEÓN, 2007).

Conforme Zadeh (1965) citado por Perez e León (2007, p. 15, tradução nossa)¹⁷ um conjunto *fuzzy* é definido como “uma classe de objetos com graus contínuos de pertinência. Este conjunto é caracterizado por uma função de pertinência (característica) a qual se atribui a cada objeto um grau de pertinência que varia entre zero e um”.

Almeida e Evsukoff (2005, p. 172) afirmam que “a teoria dos conjuntos *fuzzy* pode ser vista como uma extensão da teoria clássica de conjuntos, criada para

¹⁶ Referência citada por Weber e Klein (2003, p. 22).

¹⁷ Texto original: “Una clase de objetos con continuos grados de pertenencia. Tal conjunto es caracterizado por una función de pertenencia (característica) la cual asigna a cada objeto un grado de pertenencia que varia entre cero y uno”.

tratar graus de pertinência intermediários entre a pertinência total e a não-pertinência de elementos de um universo”.

O grau de pertinência de um elemento em relação a um conjunto *fuzzy* F e um universo de discurso U pode ser definido por uma função de pertinência, que assume como valor qualquer número pertencente ao intervalo real fechado $[0,1]$, mapeada por $\mu_F(x): U \rightarrow [0,1]$. (ALMEIDA e EVSUKOFF, 2005).

Segundo Almeida e Evsukoff (2005, p. 174), para representar conjuntos *fuzzy* são necessários basicamente da natureza e da dimensão do universo definido. Por exemplo, suponha que em um universo $U: [0,5]$, existem três conjuntos *fuzzy* denominados “Pequeno”, “Médio” e “Grande” e o conjunto A com elementos de U .

$$A : \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$$

A função de pertinência para os três conjuntos *fuzzy* é mapeada por:

$$\mu_P(x) = \begin{cases} 1, & \text{se } x < 1.5 \\ 2.5 - x, & \text{se } 1.5 \leq x \leq 2.5 \\ 0, & \text{se } x > 2.5 \end{cases}$$

$$\mu_M(x) = \begin{cases} x - 1.5, & \text{se } 1.5 \leq x \leq 2.5 \\ 3.5 - x, & \text{se } 2.5 \leq x \leq 3.5 \\ 0, & \text{se } x < 1.5 \text{ ou } x > 3.5 \end{cases}$$

$$\mu_G(x) = \begin{cases} 0, & \text{se } x < 2.5 \\ x - 2.5, & \text{se } 2.5 \leq x \leq 3.5 \\ 1, & \text{se } x > 3.5 \end{cases}$$

Para Almeida e Evsukoff (2005, p. 173), quando o universo é composto por um número pequeno de elementos, a forma de representação analítica do conjunto A para as funções de pertinência dos conjuntos *fuzzy* é realizada da seguinte forma:

$$\mu_P(x) = \{1/0, 1/0.5, 1/1, 1/1.5, 0.5/2, 0/2.5, 0/3, 0/3.5, 0/4, 0/4.5, 0/5\}$$

$$\mu_M(x) = \{0/0, 0/0.5, 0/1, 0/1.5, 0.5/2, 1/2.5, 0.5/3, 0/3.5, 0/4, 0/4.5, 0/5\}$$

$$\mu_G(x) = \{0/0, 0/0.5, 0/1, 0/1.5, 0/2, 0/2.5, 0.5/3, 1/3.5, 1/4, 1/4.5, 1/5\}$$

Todos os elementos são separados por vírgulas representados na forma “ $\mu_A(x_i)/x_i$ ”, onde o primeiro termo $\mu_A(x_i)$ representa o grau de pertinência de x_i no conjunto A e o segundo termo representa o próprio elemento x_i .

Quando o universo de discurso é contínuo ou possui uma quantidade grande de elementos discretos, a melhor forma de representação é por meio do gráfico de sua pertinência. (ALMEIDA e EVSUKOFF, 2005).

O Gráfico 2 apresenta os três conjuntos *fuzzy*, “Pequeno”, “Médio” e “Grande”, no universo U.

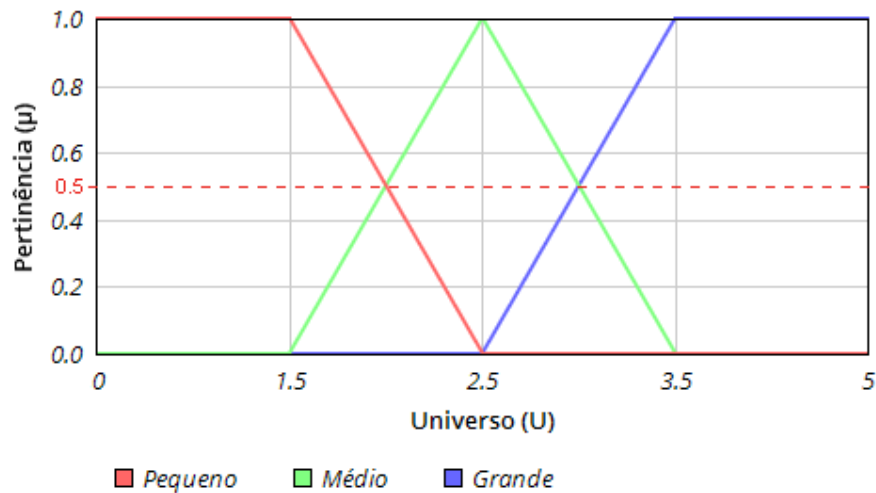


Gráfico 2: Gráfico representando os três conjuntos *fuzzy* no universo U.

4.3.1 Operações *Fuzzy*

Da mesma forma que na teoria clássica, os conjuntos *fuzzy* obedecem a propriedades e estão sujeitos a diversas operações. Zadeh (1965) citado por Perez e León (2007, p. 17) propôs diversas operações *fuzzy*. Entretanto, nesta subseção, apenas serão abordadas as operações básicas de conjuntos *fuzzy*: complemento, união e interseção.

A seguir serão apresentadas as três operações básicas, utilizando como base para a exemplificação os três conjuntos *fuzzy* apresentados na seção anterior, “Pequeno”, “Médio” e “Grande”:

- 1. Complemento:** O complemento de um conjunto *fuzzy* A no universo U pode ser denotado por $\neg A$, com uma função de pertinência definida por: $\mu_{\neg A}(x_i) = 1 - \mu_A(x_i)$. (ALMEIDA e EVSUKOFF, 2005, p. 174). Por

exemplo, a operação de complemento para o conjunto “Não Grande” é apresentada conforme o Gráfico 3:

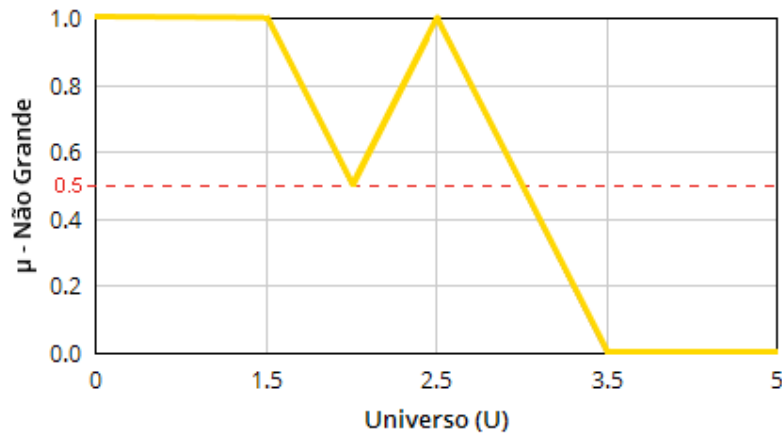


Gráfico 3: Conjunto *fuzzy* resultante da operação de complemento.

Fonte: Adaptação de Almeida e Evsukoff (2005, p. 175).

- 2. União:** A união de dois conjuntos *fuzzy* A e B do universo U pode ser denotado $A \cup B$ ou $A + B$, com uma função de pertinência definida por: $\mu_{A \cup B} = \max[\mu_A(x_i), \mu_B(x_i)]$. (ALMEIDA e EVSUKOFF, 2005). Conforme o Gráfico 4, verificamos a resultante da operação de união entre os conjuntos “Médio” e “Grande”.

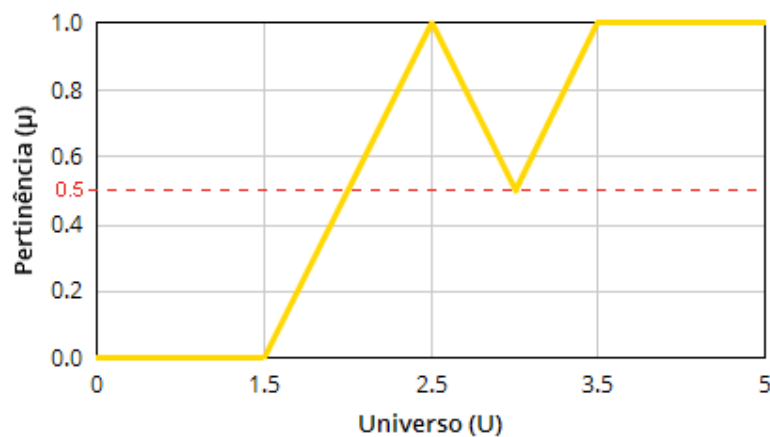


Gráfico 4: Conjunto *fuzzy* resultante da operação de união.

Fonte: Adaptação de Almeida e Evsukoff (2005, p. 176).

- 3. Interseção:** A interseção de dois conjuntos *fuzzy* A e B do universo U pode ser denotada por $A \cap B$ ou $A \cdot B$, através da função de pertinência: $\mu_{A \cap B} = \min[\mu_A(x_i), \mu_B(x_i)]$. (ALMEIDA e EVSUKOFF, 2005). O Gráfico 5, exemplifica a operação de interseção entre “Pequeno” e “Médio”.

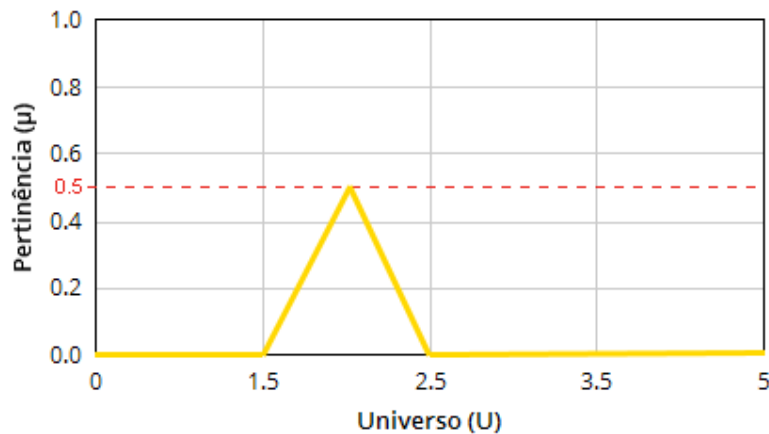


Gráfico 5: Conjunto *fuzzy* resultante da operação de interseção.

Fonte: Adaptação de Almeida e Evsukoff (2005, p. 177).

4.4 REPRESENTAÇÃO FUZZY DE CONHECIMENTO

Como verificamos anteriormente, no Capítulo 2, que para a resolução de problemas é necessário que o especialista identifique o problema, sua causa, as ações a serem tomadas e os recursos disponíveis para a sua solução, ou seja, será necessário estruturar o conhecimento sobre o problema.

De acordo com Almeida e Evsukoff (2005, p. 177), alguns problemas não permitem a definição precisa de uma solução, permitindo que seja conduzido a uma classificação ou agregação qualitativa em categorias gerais ou ainda a um conjunto de possíveis soluções.

Esta capacidade de classificar as variáveis de um problema de forma imprecisa, em termos de conceitos qualitativos em vez de quantitativos, traduz a

ideia de uma variável linguística. Fundamentalmente, o processo de representação *fuzzy* de conhecimento depende deste conceito.

4.4.1 Variável Linguística

Para Almeida e Evsukoff (2005, p. 177), o conceito de variável linguística é definido como “uma entidade utilizada para representar de modo impreciso e, portanto, linguístico, um conceito ou uma variável de um dado problema”.

A variável linguística admite como valores apenas expressões linguísticas (também chamados de termos primários), como por exemplo, “frio”, “muito grande”, “aproximadamente alto”, etc. (ALMEIDA e EVSUKOFF, 2005).

Um termo primário de uma variável linguística pode ser representado por um conjunto *fuzzy* existente no universo de discurso. Assim, em cada conjunto *fuzzy* é associado um conceito linguístico que classifica ou define um valor para a variável linguística.

De acordo com Almeida e Evsukoff (2005, p. 178), os termos primários definidos formam a sua estrutura de conhecimento, chamada de partição *fuzzy*. A forma de utilização das variáveis linguísticas depende das propriedades sintáticas e semânticas que vão reger o comportamento do sistema de conhecimento *fuzzy*.

As propriedades sintáticas definem o formato em que serão armazenadas as informações linguísticas *fuzzy*. Proporciona a criação de Base de Conhecimento contendo sentenças estruturadas, sistematizando os processos de armazenamento, busca e processamento dos dados existentes. (ALMEIDA e EVSUKOFF, 2005).

As propriedades semânticas vão especificar de que modo é extraído e processado o conhecimento, armazenado na forma de declarações condicionais *fuzzy*, ou regras de produção *fuzzy*, contida na estrutura definida pelas propriedades sintáticas. (ALMEIDA e EVSUKOFF, 2005).

4.4.2 Regras de Produção *Fuzzy*

Segundo Almeida e Evsukoff (2005, p. 179), a maneira mais comum de se armazenar informações em uma Base de Conhecimento *fuzzy* é a representação através de regras de produção *fuzzy*. Uma regra de produção pode ser composta de duas partes principais: *SE <antecedente> ENTÃO <consequente>*.

O antecedente é composto por um conjunto de condições que quando satisfeitas determinam o processamento do consequente. O consequente é uma ação executada por mecanismo de inferência *fuzzy*.

4.5 INFERÊNCIA FUZZY COM MAMDANI

De acordo com Almeida e Evsukoff (2005, p. 179), a semântica é que vai definir para o mecanismo de inferência, como serão processados os antecedentes, quais serão os indicadores de disparo das regras e quais os operadores utilizados sobre os conjuntos *fuzzy* para executar o processamento de conhecimento.

Existem vários métodos de inferência, como os modelos: de Larsen, de Tsukamoto, de Takagi e Sugeno, no entanto o modelo de Mamdani é geralmente o mais utilizado. Em nosso trabalho iremos concentrar o estudo de inferência sobre o modelo Mamdani.

O professor Ebrahim Mamdani (1974) propôs um método de inferência *fuzzy* que constitui o coração dos mecanismos de inferência utilizados atualmente nas aplicações de sistemas *fuzzy*. (PEREZ e LEÓN, 2007, p. 32).

De acordo com Almeida e Evsukoff (2005, p. 179), as regras de produção no modelo Mamdani possuem relações *fuzzy* tanto em seus antecedentes como os seus consequentes.

Devido ao grande número de sistemas convencionais serem baseados em grandezas numéricas, o modelo Mamdani possui módulos de interface com a função de realizar a conversão das variáveis de entrada em conjuntos *fuzzy* e para a saída o processo inverso. (ALMEIDA e EVSUKOFF, 2005).

A regra semântica utilizada para o processamento de inferências com o modelo Mamdani é chamada de Máx-Min. No qual utiliza as operações de união e de interseção entre conjuntos *fuzzy* através dos operadores máximo e mínimo.

No diagrama, Figura 29, a máquina de inferência recebe valores *fuzzy* provenientes do módulo de interface de entrada, processa as regras existentes na Base de Conhecimento e gera um conjunto *fuzzy* para o módulo de interface de saída. A seguir é exibido o diagrama típico do modelo Mamdani:

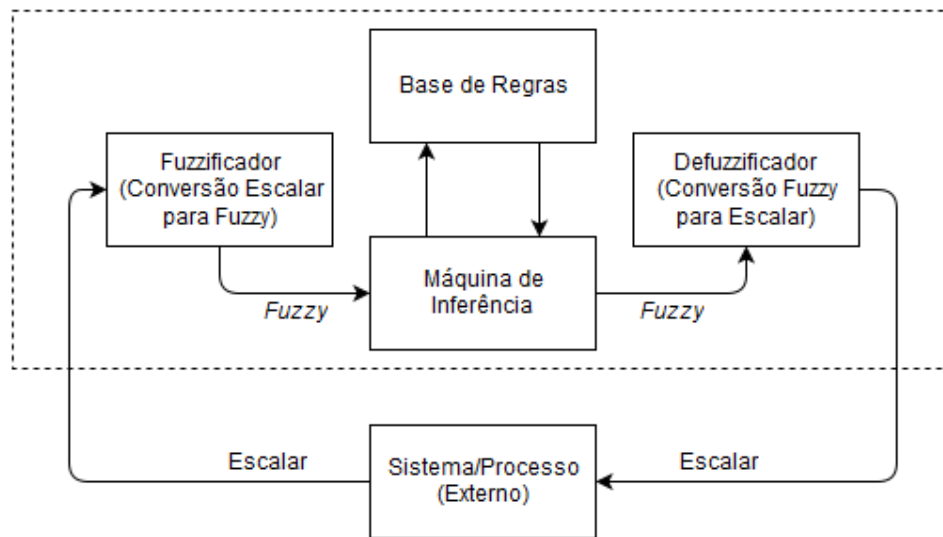


Figura 29: Diagrama típico de um modelo de inferência Mamdani.

Fonte: Adaptação de Almeida e Evsukoff (2005, p. 180); Weber e Klein (2003, p. 35).

4.5.1 Módulo de Interface de Entrada

Também denominado de fuzzificação (WEBER e KLEIN, 2003), do inglês, *fuzzification*. (ALMEIDA e EVSUKOFF, 2005). Este processo é responsável pela conversão dos dados de entrada em conjunto *fuzzy*. É mais frequente que os dados de entrada estejam no formato numérico (escalar).

De acordo com Weber e Klein (2003, p. 37), os dados de entrada devem estar normalizados para serem enquadrados dentro dos limites adequados do universo do discurso e associadas a um grau de pertinência.

4.5.2 Máquina de Inferência

É o processo de avaliação com objetivo de comparar os dados de entrada com as regras definidas na base de regras e assim obter conclusões utilizando os conjuntos *fuzzy*. (ALMEIDA e EVSUKOFF, 2005).

Almeida e Evsukoff (2005, p. 180) afirmam que os antecedentes de cada regra são processados através da interseção *fuzzy* entre os graus de pertinência das entradas nos termos primários definidos em cada uma. Este processo gera um grau de pertinência de disparo a cada regra de produção.

4.5.3 Base de Regras

A Base de Regras consiste de um Banco de Dados ou de uma SBC. Componente responsável em controlar as regras *fuzzy*, cada proposição é descrita na forma linguística associadas a um grau de pertinência. (WEBER e KLEIN, 2003).

Também estão armazenadas na Base de Regras as normalizações dos universos do discurso e as definições das funções de pertinência. A Normalização das regras torna o sistema mais fácil de ser compreendido, apesar do crescimento número de regras. (WEBER e KLEIN, 2003, p. 37).

Weber e Klein (2003) afirmam que um sistema com regras normalizadas se obtêm uma maior eficiência computacional do que a que seria obtida caso fossem usadas regras não normalizadas.

4.5.4 Módulo de Interface de Saída

Segundo Weber e Klein (2005), este processo também é denominado de defuzzificação, do inglês, *defuzzification*. (ALMEIDA e EVSUKOFF, 2005), no qual este processo transforma a conclusão obtida em uma informação de controle não

fuzzy que melhor representa a saída. (WEBER e KLEIN, 2003, p. 45). A conversão transforma informações qualitativas em informação quantitativas.

Segundo Almeida e Evsukoff (2005), a conclusão obtida ou o conjunto *fuzzy* gerado durante o processo de inferência pode então ser utilizado na sua forma qualitativa em um processo de tomada de decisão ou ser convertido no formato numérico (escalar).

Os métodos mais utilizados para defuzzificação são:

- **Método de centro de massa:** Calcula para um dado conjunto *fuzzy* de saída proveniente de uma Base de Conhecimento, a abscissa do ponto de centro de massa correspondente e a utiliza como valor escalar de saída. (ALMEIDA e EVSUKOFF, 2005).
- **Método da média dos máximos:** O valor da saída corresponde ao ponto do universo de discurso que corresponde à média dos pontos de máximo locais da função de pertinência produzida no processo de inferência. (ALMEIDA e EVSUKOFF, 2005).

4.6 LÓGICA FUZZY E OS SISTEMAS DE DECISÃO

Os modelos de inferência *fuzzy* são adequados em processos que exigem tomadas de decisão devido a sua capacidade de processar de forma eficiente as informações imprecisas e qualitativas. Conforme visto no Capítulo 2, os SAD são os sistemas que apóiam os tomadores de decisão no processo decisório.

De acordo com Almeida e Evsukoff (2005, p. 185), aplicações deste tipo são responsáveis em representar o conhecimento e a experiência existentes sobre um determinado estado do processo.

A partir do momento que os dados de entradas sobre um determinado estado é inserido no sistema, é possível inferir sua evolução temporal, as variações importantes que ocorreram ou mesmo gerar sugestões sobre as próximas ações a serem tomadas. (ALMEIDA e EVSUKOFF, 2005).

A seguir como exemplo, será apresentado um sistema simples de apoio à tomada de decisão para o caso consumo de um automóvel. O exemplo terá como

base os três conjuntos *fuzzy* “Pequeno”, “Médio” e “Grande” e que o universo do discurso seja no intervalo [0, 100].

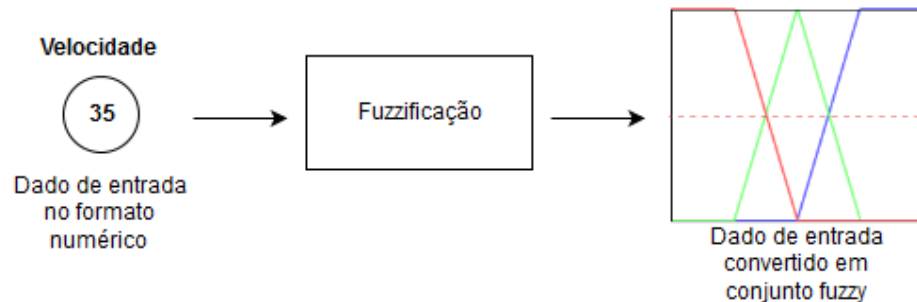


Figura 30: Processo de Fuzzificação do dado de entrada.

Na Figura 30 apresenta o processo de Fuzzificação, a entrada escalar de velocidade foi convertido no conjunto de graus de pertinência nos termos primários definidos para a linguística “Veloc”. Considere que a relação entre a velocidade e o consumo está baseada em três regras conforme a Figura 31.

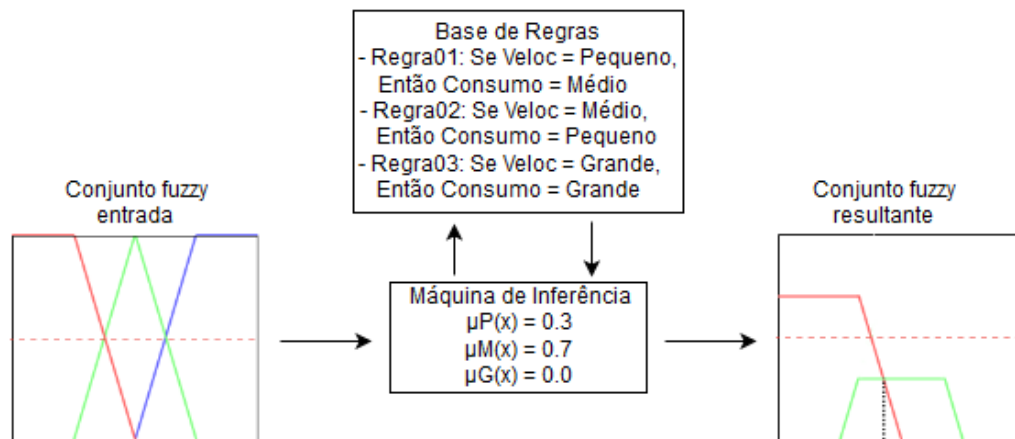


Figura 31: Processo de Inferência da relação velocidade e consumo.

No processo de Inferência, com base nas regras é montado um vetor de pertinências utilizado para limitar os conjuntos *fuzzy* de saída da variável linguística “Consumo”, segundo a Base de Conhecimento definida conforme apresentado na Figura 31.

São criadas partições *fuzzy* para estas variáveis e é definida uma regra semântica para o processo de inferência *fuzzy* a ser realizado pela técnica Max-Min de Mamdani. (ALMEIDA e EVSUKOFF, 2005).

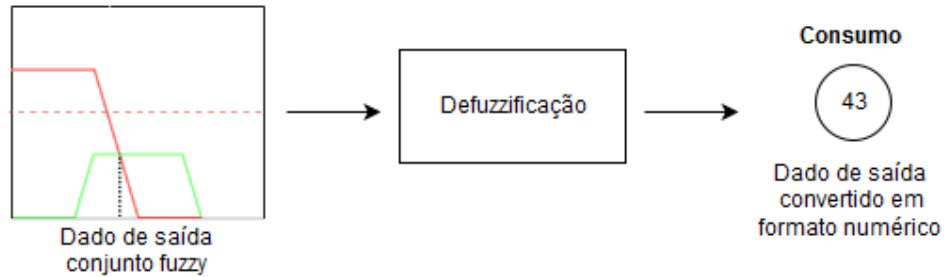


Figura 32: Processo de Defuzzificação da informação obtida.

Na Figura 32 apresenta o processo de Defuzzificação, onde ocorre a conversão do valor de saída obtido de *fuzzy* para escalar da variável “Consumo” foi obtido utilizando o método do centro de massa.

Como o processo está sendo utilizado para apoiar uma decisão, não seria necessário ocorrer o processo de defuzzificação, pois a informação de saída na forma qualitativa seria suficiente.

5 APLICAÇÃO

Neste capítulo é apresentado um caso de uso. Será descrito o cenário existente, sua estrutura, melhor solução para o problema e a conclusão obtida com sua implementação.

5.1 CENÁRIO

A empresa XPTO, pertence ao grupo das maiores empresas brasileiras do ramo farmacêutico. Possui uma conjuntura organizacional citada como referência e mesmo assim a empresa tem apresentado alguns problemas de difícil resolução.

A seguir são listados os problemas:

1. Grande dificuldade em realizar a distribuição dos medicamentos em algumas regiões, permitindo que a concorrência se intensifique nestas áreas.
2. Os setores responsáveis pela logística e expedição são compostos, em sua maioria, por colaboradores jovens e inexperientes.
3. Existem inúmeras incidências de caminhões que são despachados parcialmente carregados. Assim, um pedido que não foi despachado será enviado em outro caminhão, aumentando os custos. E se for um caminhão fretado, o custo será triplicado.
4. Cerca de 60% da frota são fretados. Há a incidência de um elevado custo sobre o transporte dos medicamentos.
5. Com o aumento dos surtos de viroses e com derivantes da gripe na população, o aumento da procura por seus medicamentos tem aumentado significativamente.
6. O SAD não tem suportado realizar buscas por todos os produtos que serão entregues para uma determinada rota e relacionar os produtos que serão despachados para que se aproveite o máximo de carga a ser transportada. Como o custo computacional do SAD é muito elevado e o processo de expedição não pode ser atrasado

ou interrompido, o processo passa a ser por decisão humana, sem existir apoio do sistema responsável.

Os gestores percebem a dificuldade de manter o controle e a diretoria estratégica concluiu o quanto se faz necessário do uso de uma nova ferramenta ou solução para auxiliar na decisão de despachar seus produtos permitindo assim o controle eficaz para o atendimento dos objetivos estratégico da empresa.

5.2 PLATAFORMA TECNOLÓGICA

Tendo como uma das principais características o grande volume de dados, a empresa XPTO possui mensalmente um gigantesco tráfego de dados e de armazenamento.

A empresa tem a necessidade de manter históricos de suas operações, dados sintéticos e analíticos, para eventuais análises e também para atender as inúmeras exigências das atuais legislações vigentes.

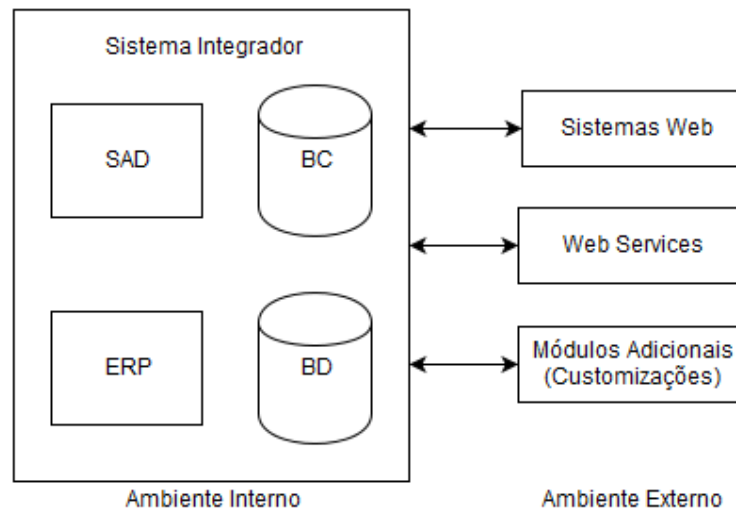


Figura 33: Plataforma tecnológica da empresa.

Na Figura 33 apresenta o ambiente interno composto por um sistema integrado de Gestão ou do inglês, *Enterprise Resource Planning* (ERP), um SAD, um Banco de dados (BD) e um Banco de conhecimento (BC).

Para atender as necessidades e não transgredir as normas de segurança estabelecidas pelo setor responsável pela segurança da informação na empresa, o ambiente interno é envolto pelo sistema integrador, de forma a impedir o acesso externo de forma eficiente e segura. Apenas no interior das dependências da empresa é possível acessar de forma direta os sistemas.

Qualquer outro tipo de sistema ou serviço deve necessariamente passar pelo sistema integrador para alcançar alguma aplicação no ambiente interno. Mesmo outra aplicação residindo no seu *datacenter* deve se autenticar para prosseguir. Qualquer aplicação interna acessa de forma direta outra aplicação no ambiente externo ao sistema integrador.

5.3 SOLUÇÃO PROPOSTA

A empresa contratou um especialista para analisar o caso. O especialista estudou o cenário, o ambiente organizacional, as peças existentes na plataforma tecnológica e a política de segurança da informação, e sugeriu que:

Em primeiro lugar, a criação de um módulo externo para realizar o devido tratamento do problema conhecido como “problema da mochila”. De acordo com Cormen (2014, p. 174), esse problema é a versão de decisão de um problema de otimização no qual queremos encher uma mochila com variados itens, desde que não excedamos um limite de peso.

Em nosso caso, o caminhão é a mochila e os pedidos são os itens. Desta forma, o sistema decidirá o conjunto de produtos relacionado ao pedido, de maneira que o cliente tenha seu pedido atendido e todos os produtos solicitados carregados no caminhão juntamente com outros pedidos de uma determinada rota.

Segundo Cormen (2014), a solução do problema da mochila é em tempo polinomial, esse tipo de problema é denominado de NP-Completo, um problema muito difícil de ser resolvido. “Um algoritmo que executa no tempo $O(n^c)$ para uma entrada n onde c é uma constante, é um algoritmo de tempo polinomial”. (CORMEN, 2014, p. 155).

O módulo externo deve ser desenvolvido de acordo com a técnica dos AG, pelo fato de ser uma técnica da IA apropriada para a resolução de problemas complexos. A comunicação com o novo módulo será via *Web Service*.

Por segundo, os produtos para serem utilizados no combate ao surto de virose e gripe, devem ser classificados no banco de conhecimento como prioritários. Esta classificação deve indicar valores quanto ao nível de prioridade numa escala de 1 a 5, por exemplo, valor igual a 3 indica uma prioridade mediana, enquanto 5 um produto de altíssima prioridade.

Em terceiro, que os colaboradores relacionados ao processo evitem que sejam tomadas decisões unilaterais após a implementação, pois somente com o uso da nova melhoria, poderá ser obtida uma métrica quanto ao ganho na operação com utilização de técnica da IA.

5.4 SOLUÇÃO EM EXECUÇÃO

O primeiro uso efetivo da solução após sua implementação no ambiente de produção foi realizado para decidir o despacho dos pedidos para a rota 05. Esta rota continha o total de 28 pedidos em aberto, ou seja, aguardando o despacho para o cliente.

Tabela 9: Lista de pedidos para a rota 05.

Número do Pedido	Peso (Kg)	Total de Prioridade	Número do Pedido	Peso (Kg)	Total de Prioridade
31208	1130	82	32108	1436	146
31225	762	60	32173	2540	124
31228	590	90	32421	986	74
31309	1048	72	32489	1960	310
31432	984	24	32490	780	160
31444	1812	132	32571	1285	80
31489	3130	110	32606	553	48
31497	1244	108	32678	1430	102
31614	4060	240	32790	1118	114
31653	760	160	32854	3090	330
31679	1146	90	33013	854	102
31725	1528	120	33244	1590	243
31966	1506	123	33262	862	78
32054	2180	320	33297	780	130

Na Tabela 09, estão listados os pedidos e respectivos peso total e valor de prioridade. Esses valores serão enviados como dados de entrada para o processamento do novo módulo customizado. O objetivo é maximizar a carga que um caminhão possa transportar e atender aos pedidos prioritários.

Para esse despacho foi designado um caminhão de eixo isolado com 4 (quatro) pneumáticos, com carga máxima de 10.750 *Kg* segundo a Resolução 12/98 e 104/99 do DNIT.

O cliente para acesso ao serviço de *web service* da nova solução será requisitada pelo SAD. Substituindo desta forma o processo nativo de elevado custo computacional pelo processo que utiliza AG.

O serviço dispõe de 7 parâmetros de entrada:

1. Vetor com os números dos pedidos.
2. Vetor com os pesos.
3. Vetor com os valores de prioridade.
4. Carga máxima do caminhão.
5. Taxa de mutação.
6. Número de gerações.
7. Tamanho da população. (Número de itens do vetor de entrada).

Ao ser executado o serviço, imediatamente a aplicação retornou com os pedidos que deveriam ser enviados. O retorno da aplicação será de um vetor com os pedidos selecionados quanto a sua prioridade.

```
Melhor da geracao #16 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #17 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #18 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #19 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #20 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #21 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #22 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #23 = 0000000101000100011010001101 com avaliação 1581.0
Melhor da geracao #24 = 0011000001000100011000001101 com avaliação 1587.0
Indivíduo escolhido = 0011000001000100011000001101
31228, 31309, 31653, 32054, 32489, 32490, 33013, 33244, 33297
```

Figura 34: Retorno da aplicação com os pedidos selecionados.

Verificando o retorno, temos um vetor com 9 pedidos. Somando todos os pesos referentes aos pedidos selecionados, temos um total de 10.542 *Kg* de caixas.

Percebemos que o caminhão com capacidade total de 10.750 Kg estará totalmente carregado para entregas na rota 05.

Para o segundo teste de uso da nova ferramenta, foi realizado para a rota 02, com o total de 53 pedidos ainda não atendidos. Desta vez a quantidade de gerações foi alterada para 100. Mesmo com o aumento de indivíduos e de gerações, a aplicação não apresentou aumento no custo computacional, retornando com os pedidos a serem enviados de forma imediata.

A aplicação, o serviço de *web service* e seu cliente foram desenvolvidos em Java. Os códigos fontes se encontram no apêndice deste trabalho.

5.5 BENEFÍCIOS ADQUIRIDOS

No mês seguinte a implantação, uma medição foi realizada para apurar e foram constatadas as seguintes mudanças:

- Os pedidos não atendidos tinham somente 03 dias desde a data de sua solicitação, não apresentando atrasos quando em comparação dos meses anteriores, onde haviam pedidos com até 10 dias de atraso esperando para serem despachados.
- A quantidade de caminhões despachados foi menor em 45% com base na média mensal de 355 ao longo de 12 meses antes da implantação da nova solução.
- A quantidade de pedidos atendidos neste último mês foi superior aos demais devido ao surto de virose e gripe. Devido a este fato, foi possível perceber o quanto a implementação da nova solução é responsável em ajudar na decisão do despacho de forma eficiente dos caminhões.

A diretoria estratégica da empresa aguarda os resultados do próximo mês para decidir em quanto irá reduzir a frota fretada. Conseqüentemente, essa ação irá diminuir consideravelmente os custos de transporte sobre os produtos, tornando-os mais competitivos à frente dos concorrentes.

6 CONCLUSÃO

Ao longo deste trabalho, foram realizados estudos concentrados em duas técnicas da IA, o AG e a LF. Existem também outras técnicas tradicionais e da IA de resolução de problemas que podem ser aplicadas em várias situações para se obter uma solução com maior índice de eficiência.

Verificamos que os SAD que não possuem estas ferramentas integradas ajudam a apoiar uma decisão desde que o problema a ser solucionado não seja um problema muito difícil de ser resolvido. O uso da ferramenta correta proporcionará o melhor desempenho.

Com o AG utilizado no Capítulo 5, percebemos que o tempo de execução foi excelente, devido ao grau de complexidade do método *executa* da classe *AlgGen* possuir notação $O(n^2)$. Como este método é o pilar central do AG, sua execução é ágil quando estabelecemos um número n de gerações.

No AG, também foi utilizada a estratégia do elitismo no qual preservamos o melhor indivíduo evitando assim descartar uma possível solução do problema ao longo das gerações.

Quanto a LF, não houve a sua implementação na prática, apenas no final do Capítulo 4, há um exemplo bem simples de como seria a sua associação ao SAD. Uma aplicação na qual converteria um dado numérico em um conjunto *fuzzy*, buscaria sua correlação na Base de Conhecimento, retornaria uma informação na forma qualitativa para o apoio a decisão.

As técnicas estudadas ao serem aplicadas ao SAD ajudam a aperfeiçoar um ou mais processos. No SAD, o resultado final é o conhecimento. A decisão não será realizada pelo sistema, será realizada pelo tomador de decisão. Então qualquer que seja a técnica aplicada ao SAD para torná-lo mais preciso é de grande importância a uma organização.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho propõe as seguintes possibilidades de trabalhos futuros:

- O desenvolvimento de uma aplicação para incluir regras de produção *fuzzy* na base de conhecimento, utilizando expressões linguísticas para associar registros transacionais. Podemos citar como exemplo a sazonalidade de produtos em uma rede de varejo, permitindo que o SAD simule de forma mais precisa uma previsão de venda para este produto em uma determinada época do ano e planejando uma aquisição tolerante para evitar perdas.
- A utilização de técnicas de meta-heurística como: Pesquisa Tabu, Colônia de Formigas e Enxame de Partículas. De forma a realizar um estudo consistente dessas técnicas em combinação com o SAD para obter o melhor desempenho.
- Desenvolver uma aplicação utilizando a técnica de Redes Neurais para que de forma automatizada realize a organização das regras existentes em uma Base de Conhecimento, realizar sua devida normalização e a descoberta de novas combinações de regras nos bancos de dados transacionais.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTS, B. et al. **Biologia Molecular da Célula**. Tradução de Ana Letícia de Souza Vanz; Anne D. Villela, *et al.* 5ª. ed. Porto Alegre: Artmed, 2010. ISBN 978-85-363-2066-3.

ALMEIDA, P. E. M. D.; EVSUKOFF, A. G. Sistemas Fuzzy. In: REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações**. Barueri: Manole, 2005. Cap. 7, p. 169-200.

BALTZAN, P.; PHILLIPS, A. **Sistemas de Informação**. Tradução de Rodrigo Dubal. Porto Alegre: AMGH, 2012.

BELFIORE, P.; FÁVERO, L. P. **Pesquisa Operacional para Cursos de Engenharia**. Rio de Janeiro: Elsevier, 2013.

BROWNE, J. **"A Origem das Espécies" de Darwin**: uma biografia. Tradução de Maria Luiza X. de A. Borges. Rio de Janeiro: Zahar, 2007.

CARVALHO, A. C. P. D. L. F. D.; BRAGA, A. D. P.; LUDERMIR, T. B. Computação Evolutiva. In: REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações**. Barueri: Manole, 2005. Cap. 9, p. 225-246.

CORMEN, T. H. **Desmiticando algoritmos**. Tradução de Arlete Simille Marques. 1ª. ed. Rio de Janeiro: Elsevier, 2014. ISBN 978-85-352-7177-5.

COX, M. M.; DOUDNA, J. A.; O'DONNELL, M. **Biologia Molecular**: Princípios e Técnicas. Tradução de Gaby Renard e Jocelei Maria Chies. Porto Alegre: Artmed, 2012.

CRUZ, T. **Gerência do Conhecimento**. 2ª. ed. Rio de Janeiro: E-papers, 2007. ISBN 978-85-7650-129-9.

HENDERSON, M. **50 Ideias de Genética**: que precisa saber mesmo. Tradução de Isabel Ferro Mealha e Eduarda Melo Cabrita. Alfragide: Leya, 2011.

JUNIOR, C. C. **Sistemas Integrados de Gestão - ERP**: Uma abordagem gerencial. 3ª. ed. Curitiba: Ibpex, 2008.

LACERDA, E. G. M. D.; CARVALHO, A. C. P. L. F. D. Introdução aos Algoritmos Genéticos. In: GALVÃO, C. D. O.; VALENÇA, M. J. D. S. **Sistemas Inteligentes**: Aplicações a Recursos Hídricos e Ciências Ambientais. Porto Alegre: Editora da UFRGS, 1999. Cap. 3, p. 87-148. ISBN 85-7025-527-6.

LAUDON, K.; LAUDON, J. **Sistemas de Informações Gerenciais**. Tradução de Luciana do Amaral Teixeira. 9ª. ed. São Paulo: Pearson, 2011.

LINDEN, R. **Algoritmos Genéticos**. 2ª. ed. Rio de Janeiro: Brasport, 2008. ISBN 978-85-7452-373-6.

LINDEN, R. Algoritmos Genéticos em Java. **MundoJ**, Curitiba, n. 59, p. 50-64, Maio/Junho 2013.

LOH, S. **31 tipos de sistemas de informação - 31 maneiras de a tecnologia da informação ajudar as organizações**. 1ª. ed. Porto Alegre: [s.n.], 2014. 142 p. ISBN 978-85-916683-0-4.

MAIA, W. D. A. **Percepção & Inteligência Artificial - Conceitos, Considerações e Arquitetura**. 1ª. ed. São Paulo: Biblioteca24horas, 2012.

O'BRIEN, J. A.; MARAKAS, G. M. **Administração de Sistemas de Informação - Uma introdução**. Tradução de Edgar Amorim Junior e José Hailton Simões. Tradução da 13ª. ed. São Paulo: McGraw-Hill Brasil, 2007. ISBN 85-86804-77-0.

PEREZ, I.; LEÓN, B. **Lógica Difusa para Principiantes**: Teoría y Práctica. Caracas: Universidad Valle del Monboy; Universidad Católica Andrés Bello, 2007.

RAINER, R. K.; CEGIELSKI, C. G. **Introdução a Sistemas de Informação**. Tradução de Daniel Vieira. 3ª. ed. Rio de Janeiro: Elsevier, 2012. ISBN 978-0470-47352-8.

REZENDE, D. A. **Engenharia de Software e Sistemas de Informação**. 3ª. ed. Rio de Janeiro: Brasport, 2005.

REZENDE, S. O.; PUGLIESI, J. B.; VAREJÃO, F. M. Sistemas Baseados em Conhecimento. In: REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações**. Barueri: Manole, 2005. Cap. 2, p. 13-49.

ROBINSON, T. R. **Genética para leigos**. Tradução de Gustavo Alves Bezerra. 1ª. ed. Rio de Janeiro: Alta Books, 2015. ISBN 978-85-7608-916-2.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. Tradução de Regina Célia Simile. Tradução da 3ª. ed. Rio de Janeiro: Elsevier, 2013. ISBN 978-85-352-3701-6.

TAKEUCHI, H.; NONAKA, I. Criação e Dialética do Conhecimento. In: TAKEUCHI, H.; NONAKA, I. **Gestão do Conhecimento**. Tradução de Ana Thorell. 1ª. ed. São Paulo: Bookman, 2009. Cap. 1, p. 17-38.

TOSCANI, L. V.; VELOSO, P. A. S. **Complexidade de Algoritmos: análise, projeto e métodos**. 3º. ed. Porto Alegre: Bookman, 2012. ISBN 978-85-407-0139-7.

TURBAN, E.; VOLONINO, L. **Tecnologia da Informação para Gestão**. Tradução de Aline Evers. 8ª. ed. Porto Alegre: Bookman, 2013.

TURBAN, E.; WETHERBE, J. C.; MCLEAN, E. **Tecnologia da Informação para Gestão**. 7ª. ed. Porto Alegre: Bookman, 2007.

WATSON, J. D. et al. **Biologia Molecular do Gene**. Tradução de Andréia Escosteguy Vargas; Luciane M. P. Passaglia e Rivo Fischer. 7ª. ed. Porto Alegre: Artmed, 2015.

WEBER, L.; KLEIN, P. A. T. **Aplicação da Lógica Fuzzy em Software e Hardware.**
Canoas: ULBRA, 2003.

APÊNDICE

Interface do serviço WSAG
<pre> package AG; import javax.jws.WebService; import javax.jws.WebMethod; @WebService public interface WSAGInterface { @WebMethod String[] agenteGen(String[] pedidos, double[] pesos, double[] valores, double cargaMax, double txMutacao, int numGeracoes, int tamPopulacao); } </pre>

Serviço WSAG
<pre> package AG; import javax.jws.WebService; import AG.WSAGInterface; @WebService(endpointInterface = "AG.WSAGInterface") public class WSAGServer implements WSAGInterface { public String[] agenteGen(String[] pedidos, double[] pesos, double[] valores, double cargaMax, double txMutacao, int numGeracoes, int tamPopulacao) { String[] pedSelec = new String[pedidos.length]; Indivíduo melhor; String escolhidos; if (pedidos.length != pesos.length && pedidos.length != valores.length) { return null; } AlgGen AG = new AlgGen(pesos, valores, cargaMax); melhor = AG.executa(txMutacao, numGeracoes, tamPopulacao); escolhidos = melhor.getIndivíduo().toString(); for (int i = 0; i < escolhidos.length(); i++) { if (escolhidos.charAt(i) == '1') { pedSelec[i] = pedidos[i]; } } return pedSelec; } } </pre>

Publicar o serviço WSAG

```

package AG;

import javax.xml.ws.Endpoint;

public class WSAGPublish {

    public static void main(String[] args)
    {
        Endpoint.publish("http://127.0.0.1:9876/alggen", new WSAGServer());
    }
}

```

Cliente para acesso ao serviço WSAG

```

package AG;

import java.net.URL;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;

public class ClienteWSAG {

    public String[] acessaWSAG(String[] pedidos, double[] pesos,
                               double[] valores, double cargaMax,
                               double txMutacao, int numGeracoes,
                               int tamPopulacao) throws Exception {
        URL url = new URL("http://127.0.0.1:9876/alggen?wsdl");
        QName qname = new QName("http://AG/", "WSAGServerService");
        Service ws = Service.create(url, qname);
        WSAGServer AG = ws.getPort(WSAGServer.class);
        String[] pedSelec = AG.agenteGen(pedidos, pesos, valores,
                                         cargaMax, txMutacao,
                                         numGeracoes, tamPopulacao);

        return pedSelec;
    }
}

```

Web Service - http://127.0.0.1:9876/alggen?wsdl

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS
RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
<!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS
RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e51e.
-->
- <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://AG/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://AG/"
name="WSAGServerService">
- <types>
- <xsd:schema>

```

```

<xsd:import namespace="http://AG/"
schemaLocation="http://localhost:9876/alggen?xsd=1" />
</xsd:schema>
</types>
- <message name="agenteGen">
  <part name="parameters" element="tns:agenteGen" />
</message>
- <message name="agenteGenResponse">
  <part name="parameters" element="tns:agenteGenResponse" />
</message>
- <portType name="WSAGInterface">
- <operation name="agenteGen">
  <input wsam:Action="http://AG/WSAGInterface/agenteGenRequest"
message="tns:agenteGen" />
  <output wsam:Action="http://AG/WSAGInterface/agenteGenResponse"
message="tns:agenteGenResponse" />
</operation>
</portType>
- <binding name="WSAGServerPortBinding" type="tns:WSAGInterface">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"
/>
- <operation name="agenteGen">
  <soap:operation soapAction="" />
- <input>
  <soap:body use="literal" />
</input>
- <output>
  <soap:body use="literal" />
</output>
</operation>
</binding>
- <service name="WSAGServerService">
- <port name="WSAGServerPort" binding="tns:WSAGServerPortBinding">
  <soap:address location="http://localhost:9876/alggen" />
</port>
</service>
</definitions>

```

Interface da classe AlgGen

```

package AG;

public interface IAlgGen {
    void inicializaPopulacao(int tamPopulacao);
    Individo executa(double txMutacao, int numGeracoes, int tamPopulacao);
}

```

Classe AlgGen

```

package AG;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;

/*
 * Adaptado de Linden (2013).
 */

```



```

public class AlgGen implements IAlgGen {
    private List<Individuo> populacao = new ArrayList<Individuo>();
    private double[] pesos;
    private double[] valores;
    double cargaMax = 0;

    private AlgGen() {
        populacao = new ArrayList<Individuo>();
    }

    public AlgGen(double[] pesos, double[] valores, double cargaMax) {
        this();
        this.pesos = new double[pesos.length];
        this.valores = new double[valores.length];
        this.cargaMax = cargaMax;

        if (pesos.length != valores.length) {
            if (pesos.length > valores.length) {
                this.valores = new double[pesos.length];
                for (int i = 0; i < (pesos.length - valores.length); i++) {
                    this.valores[valores.length] = 0;
                }
                for (int j = 0; j < valores.length; j++) {
                    this.valores[j] = valores[j];
                }
            } else {
                this.pesos = new double[valores.length];
                for (int i = 0; i < (valores.length - pesos.length); i++) {
                    this.pesos[pesos.length] = 0;
                }
                for (int j = 0; j < pesos.length; j++) {
                    this.pesos[j] = pesos[j];
                }
            }
        }

        for (int i = 0; i < pesos.length; i++) {
            this.pesos[i] = pesos[i];
            this.valores[i] = valores[i];
        }
    }

    public void inicializaPopulacao(int tamPopulacao) {
        this.populacao.clear();
        for(int i = 0; i < tamPopulacao; i++) {
            this.populacao.add(new Individuo(pesos, valores, cargaMax));
        }
    }

    private double somaAvaliacoes() {
        double retorno = 0;
        Iterator<Individuo> it = this.populacao.iterator();

        while(it.hasNext()) {
            Individuo prox = it.next();
            retorno += prox.avaliacao();
        }

        return(retorno);
    }
}

```

```

    }

    private int selecionaPai(double somaAvaliacoes) {
        int retorno = -1;
        double valorSorteado = Math.random() * somaAvaliacoes;
        double soma = 0;
        Iterator<Individuo> it = this.populacao.iterator();

        do {
            soma += it.next().avaliacao();
            retorno++;
        } while ((it.hasNext()) && (soma < valorSorteado));

        return(retorno);
    }

    public Individuo executa(double txMutacao, int numGeracoes,
                            int tamPopulacao) {
        Individuo melhor = null, elite = null;
        Individuo[] filhos;
        List<Individuo> novaPopulacao = new ArrayList<Individuo>();
        inicializaPopulacao(tamPopulacao);

        for (int gerAtual = 0; gerAtual < numGeracoes; gerAtual++) {
            double somaAvaliacoes = this.somaAvaliacoes();
            novaPopulacao.clear();

            for (int indGerados = 0; indGerados < tamPopulacao; indGerados += 2){
                int pai1 = this.selecionaPai(somaAvaliacoes);
                int pai2 = this.selecionaPai(somaAvaliacoes);
                filhos = populacao.get(pai1).crossover(populacao.get(pai2));
                novaPopulacao.add(filhos[0].mutacao());
                novaPopulacao.add(filhos[1].mutacao());
            }

            elite = populacao.get(populacao.size() - 1);
            populacao.clear();
            populacao.addAll(novaPopulacao);
            populacao.add(elite);
            Collections.sort(populacao);
            melhor = populacao.get(populacao.size() - 1);
        }

        return melhor;
    }
}

```

Interface da classe Individuo

```

package AG;

public interface IIndividuo {
    Individuo[] crossover (Individuo o);
    Individuo mutacao ();
    double avaliacao();
}

```

Classe Individuo

```

package AG;

import java.lang.reflect.Array;

/*
 * Adaptado de Linden (2013).
 */
public class Indivíduo implements IIndivíduo, Comparable<Indivíduo> {
    private StringBuffer indivíduo = new StringBuffer("");
    private double txMutacao;
    private double[] pesos;
    private double[] valores;
    private double cargaMax = 0;

    public Indivíduo() {
        indivíduo = new StringBuffer("");
        txMutacao = 0.1;
    }

    public Indivíduo(double[] pesos, double[] valores, double totalPeso) {
        super();
        this.pesos = pesos;
        this.valores = valores;
        this.cargaMax = totalPeso;

        for (int i = 1; i <= this.pesos.length; i++) {
            if (Math.random() < 0.5) {
                indivíduo.append("0");
            } else {
                indivíduo.append("1");
            }
        }
    }

    public Indivíduo[] crossover (Indivíduo o) {
        Class<? extends Indivíduo> c = this.getClass();
        Indivíduo[] filhos = null;
        filhos = (Indivíduo[]) Array.newInstance(c, 2);
        int posCorte = ((int) Math.round(Math.random() * this.indivíduo.length()
- 1) + 1;
        String filho01, filho02;

        try {
            for (int i = 0; i <= 1; i++) {
                filhos[i] = new Indivíduo();
                filhos[i].setPesos(pesos);
                filhos[i].setValores(valores);
                filhos[i].setTxMutacao(txMutacao);
                filhos[i].setCargaMax(cargaMax);
            }

            filho01 = o.getIndivíduo().substring(0, posCorte) +
this.getIndivíduo().substring(posCorte);
            filho02 = this.getIndivíduo().substring(0, posCorte) +
o.getIndivíduo().substring(posCorte);
            filhos[0].setIndivíduo(new StringBuffer(filho01));
            filhos[1].setIndivíduo(new StringBuffer(filho02));
        } catch (Exception ex) {
            System.out.println("Exceção: " + ex.getMessage());

```

```

        filhos = null;
    }

    return (filhos);
}

public Indivíduo mutacao () {
    Indivíduo filho = null;
    StringBuffer resultado = new StringBuffer();

    try {
        filho = new Indivíduo();
        for (int i = 0; i < this.indivíduo.length(); i++) {
            if (Math.random() < this.txMutacao) {
                if (this.indivíduo.charAt(i) == '1') {
                    resultado.append('0');
                } else {
                    resultado.append('1');
                }
            } else {
                resultado.append(this.indivíduo.charAt(i));
            }
        }

        filho.setIndivíduo(new StringBuffer(resultado));
        filho.setPesos(pesos);
        filho.setValores(valores);
        filho.setTxMutacao(txMutacao);
        filho.setCargaMax(cargaMax);

    } catch (Exception ex) {
        filho = null;
    }
    return(filho);
}

public double avaliacao() {
    double retorno = 0;
    double somaPesos = 0;
    String c = this.getIndivíduo().toString();

    for (int i = 0; i < c.length(); i++) {
        if (c.charAt(i) == '1') {
            retorno += getValores()[i];
            somaPesos += getPesos()[i];
        }
    }

    if (somaPesos > this.getCargaMax()) {
        retorno = 1;
    }

    return(retorno);
}

@Override
public boolean equals(Object o) {
    if (!this.getClass().equals(o.getClass())) {

```

```
        return false;
    }

    return(this.getIndividuo().equals((Individuo)o));
}

public int compareTo(Individuo o) {
    return ((new Double(this.avaliacao())).compareTo(o.avaliacao()));
}

public String toString() {
    return (this.individuo.toString());
}

/*
 * Getters and Setters
 */
public double[] getValores() {
    return valores;
}

public void setValores(double[] valores) {
    this.valores = valores;
}

public double[] getPesos() {
    return pesos;
}

public void setPesos(double[] pesos) {
    this.pesos = pesos;
}

public double getTxMutacao() {
    return txMutacao;
}

public void setTxMutacao(double txMutacao) {
    this.txMutacao = txMutacao;
}

public double getCargaMax() {
    return cargaMax;
}

public void setCargaMax(double cargaMax) {
    this.cargaMax = cargaMax;
}

public StringBuffer getIndividuo() {
    return individuo;
}

public void setIndividuo(StringBuffer individuo) {
    this.individuo = individuo;
}
}
```