



Programa Interdisciplinar de Pós-Graduação em

# **Computação Aplicada**

**Mestrado Acadêmico**

André da Silva Davi

Uma Aplicação de Metaheurística na Abordagem  
Do Problema de *Layout* de Armazém

São Leopoldo, 2017

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS  
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO  
EM COMPUTAÇÃO APLICADA  
NÍVEL MESTRADO

ANDRÉ DA SILVA DAVI

**UMA APLICAÇÃO DE METAHEURISTICAS NA ABORDAGEM  
DO PROBLEMA DE *LAYOUT* DE ARMAZÉM**

São Leopoldo  
2017

**UMA APLICAÇÃO DE METAHEURISTICA NA ABORDAGEM  
DO PROBLEMA DE *LAYOUT* DE ARMAZÉM**

Dissertação apresentada como requisito parcial para a  
obtenção do título de Mestre, pelo Programa  
Interdisciplinar de Pós-Graduação em Computação  
Aplicada da Universidade do Vale do Rio dos Sinos  
– UNISINOS

Orientador: Dr. José Vicente Canto dos Santos

São Leopoldo

2017

D245 Davi, André da Silva

Uma aplicação de metaheurísticas na abordagem do problema de *layout* de armazém / por André da Silva Davi. — 2017.

90 f. : il., 30 cm.

Dissertação (Mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2016.

Orientação: Prof. Dr. José Vicente Canto dos Santos, Unidade Acadêmica de Pesquisa e Pós-Graduação.

1. Algoritmo genético. 2. Centro de Distribuição. 3. Metaheurísticas. I. Davi, André da Silva. II. Santos, José Vicente Canto dos. IV. Título.

CDU 658.7

André da Silva Davi

Uma aplicação de metaheurísticas na abordagem do problema de *layout* de armazém

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 12 de setembro de 2017.

BANCA EXAMINADORA

---

Leonardo Dagnino Chiwiacowsky – Universidade de Caxias do Sul (UCS)

---

Prof. Dr. Sandro José Rigo – UNISINOS

---

Prof. Dr. José Vicente Canto dos Santos – UNISINOS

Prof. Dr. José Vicente Canto dos Santos (Orientador)

Visto e permitida a impressão  
São Leopoldo,

Prof. Dr. Sandro José Rigo  
Coordenador PPG em Computação Aplicada

## RESUMO

Neste trabalho foi desenvolvido um modelo computacional para a otimização de *layout* de um Armazém. Além da abordagem do Problema de *Layout* de Armazém, também é abordado o Problema de Família de Partes. Para o desenvolvimento do modelo foi aplicada a metaheurística Algoritmo Genético. O objetivo do estudo é projetar a configuração de um armazém que otimize a alocação de mercadorias nas prateleiras tal que proporcione a minimização da movimentação das mesmas durante a seleção de pedidos, pois a operação e a gerência são partes essenciais das operações e serviços realizados. Para isto, as variáveis de decisão são: a distância absoluta da localização da mercadoria e o número de pedidos por dia. O resultado deste trabalho é a geração de um *layout* capaz de comportar as mercadorias de acordo com as necessidades de alocação, realizando uma otimização de aproximadamente 15%.

**Palavras-Chave:** Algoritmo Genético; Centro de Distribuição; Metaheurísticas; *Layout* de Armazém

## **ABSTRACT**

In this work a computational model was developed for a warehouse layout optimization. In addition to the Warehouse Layout Problem approach, the Part Family Problem is also addressed. For the development of the model was applied the metaheuristic Genetic Algorithm. The objective of the study is to design the configuration of a warehouse that optimizes an allocation of goods on the shelves that provides a minimization of the warehouse's movement during order selection, operation and management with essential uses of the operations and services performed. The decision variables are: the absolute distance of the location of the merchandise and the number of requests per day. The result is a set of new layouts, according to the conditions of service and the realization of an optimization of approximately 15%.

**Keywords:** Genetic Algorithm; Distribution Center; Metaheuristics; Warehouse Layout.

## LISTA DE FIGURAS

Figura 1: Metodologia adotada na presente dissertação .....	14
Figura 2: Evolução do pensamento logístico.....	18
Figura 3: Sistema Logístico.....	19
Figura 4: Fluxo de mercadorias em um centro de distribuição tradicional .....	23
Figura 5: Percentual dos tempos típicos no processo de seleção de pedidos .....	25
Figura 6: Matriz de incidência antes e depois do agrupamento .....	29
Figura 7: Classificação do <i>Bin Packing</i> de acordo com as dimensões.....	30
Figura 8: Etapas do AG proposto por Holland.....	36
Figura 9: Pseudocódigo do Algoritmo Genético .....	37
Figura 10: Representação de cromossomos binários e reais. ....	39
Figura 11: Representação da seleção pelo método de torneio.....	40
Figura 12: Simulação da seleção por roleta.....	41
Figura 13: Cruzamento com ponto de corte único. ....	42
Figura 14: Cruzamento com ponto de corte duplo. ....	42
Figura 15: Cruzamento com pontos aleatórios .....	43
Figura 16: Cruzamento utilizando PMX .....	43
Figura 17: Mutação em (a) apenas um gene e no cromossomo (b) todos os genes.....	44
Figura 18: <i>Layout</i> tradicional e espinha de peixe .....	48
Figura 19: Seleção manual de pedidos .....	49
Figura 20: Espaços para a movimentação das mercadorias .....	50
Figura 21: Forma de armazenamento e sistema de identificação de endereço.....	51
Figura 22: Cubagem das mercadorias .....	52
Figura 23: Módulos do protótipo.....	56
Figura 24: Pseudocódigo para a criação das FPs.....	57



Figura 25: Distribuição das mercadorias nas estantes .....	57
Figura 26: Módulo de otimização.....	58
Figura 27: Fluxo do processo de otimização .....	59
Figura 28: Representação da arquitetura Cliente-Servidor.....	61
Figura 29: Arquitetura da aplicação .....	61
Figura 30: Gerenciamento das requisições dos Session Beans pelo contêiner.....	64
Figura 31: Arquitetura do JPA – <i>Java Persistence Api</i> .....	66
Figura 32: Fluxo de criação da base dados.....	67
Figura 33: Console de administração do servidor Wildfly.....	68
Figura 34: <i>Template</i> aplicação geral da aplicação.....	69
Figura 35: Ciclo de vida do JSF .....	70
Figura 36: Apresentação do armazém .....	71
Figura 37: Número de gerações x tempo e distância.....	72
Figura 38: Tamanho da população x distância e tempo .....	73
Figura 39: Taxa de mortalidade x distância e tempo.....	74
Figura 40: Histograma e distribuição normal da instância i-100 .....	75
Figura 41: Histograma e distribuição normal da instância i-300 .....	75
Figura 42: Histograma e distribuição normal da instância i-500 .....	76

## LISTA DE TABELAS

Tabela 1: Tempo para a solução em relação ao conjunto de entrada. ....	35
Tabela 2: Análise do vocabulário do AG x problema computacional.....	36
Tabela 3: Resultado das pesquisas.....	46
Tabela 4: Trabalhos retornados para “ <i>warehouse layout problem</i> ” .....	47
Tabela 5: Dados de entrada referente as mercadorias .....	56
Tabela 6: Resumo dos resultados .....	74
Tabela 7: Processador e memória utilizado pelos trabalhos.....	76
Tabela 8: Comparação entre os trabalhos.....	77

## LISTA DE SIGLAS

AG	Algoritmo Genético
CD	Centro de Distribuição
PIPCA	Programa Interdisciplinar de Pós-Graduação em Computação Aplicada
UNISINOS	Universidade do Vale do Rio dos Sinos
WMS	<i>Warehouse Management System</i>

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>1.1 Motivação .....</b>	<b>11</b>
<b>1.2 Problema .....</b>	<b>12</b>
<b>1.3 Objetivos.....</b>	<b>13</b>
<b>1.4 Metodologia.....</b>	<b>Erro! Indicador não definido.</b>
<b>1.5 Organização do texto.....</b>	<b>14</b>
<b>2 LOGÍSTICA .....</b>	<b>17</b>
<b>2.1 Conceito e Evolução da Logística.....</b>	<b>17</b>
<b>2.2 Logística de Distribuição .....</b>	<b>19</b>
<b>2.3 Armazenagem .....</b>	<b>20</b>
<b>2.4 Layout de Armazém.....</b>	<b>21</b>
<b>2.5 Centro de Distribuição .....</b>	<b>22</b>
<b>2.6 Seleção de pedidos .....</b>	<b>23</b>
<b>2.7 Warehouse Management System .....</b>	<b>25</b>
<b>3 TÉCNICAS E CONCEITOS PARA ABORDAR O PROBLEMA .....</b>	<b>27</b>
<b>3.1 Tecnologia de Grupo .....</b>	<b>27</b>
<b>3.2 Bin Packing.....</b>	<b>29</b>
3.2.1 Uma dimensão .....	30
3.2.2 Duas dimensões .....	30
3.2.3 Três dimensões .....	31
<b>3.3 Problema de Layout de Armazém .....</b>	<b>32</b>
<b>3.4 Metaheurísticas.....</b>	<b>33</b>
3.4.1 Algoritmos Genéticos .....	35
3.4.1.1 População inicial .....	37
3.4.1.2 Cromossomo.....	38
3.4.1.3 Função Objetivo .....	39
3.4.1.4 Processo de Seleção.....	39
3.4.1.5 Operadores de Cruzamento, Mutação e Epidemia .....	41
3.4.1.6 Critério de parada .....	45

<b>3.5 Trabalhos relacionados .....</b>	<b>45</b>
<b>4 MODELO .....</b>	<b>49</b>
<b>4.1 Apresentação do Ambiente .....</b>	<b>49</b>
<b>4.2 Definição das regras do Modelo .....</b>	<b>52</b>
<b>4.3 Formulação Matemática .....</b>	<b>54</b>
<b>4.4 Arquitetura da aplicação .....</b>	<b>55</b>
<b>4.5 Tecnologias aplicadas .....</b>	<b>59</b>
4.5.1 Engenharia de <i>Software</i> .....	60
4.5.2 Arquitetura da aplicação quanto a tecnologia.....	60
4.5.3 Linguagem de programação .....	62
4.5.4 Enterprise JavaBeans .....	63
4.5.5 Banco de dados .....	64
4.5.6 Camada de persistência .....	65
4.5.7 Servidor de Aplicação Web .....	67
4.5.8 Interface .....	68
<b>5 TESTES E RESULTADOS .....</b>	<b>71</b>
<b>5.1 Experimentos e Calibração.....</b>	<b>71</b>
<b>5.2 Comparação com outros resultados.....</b>	<b>76</b>
<b>6 CONCLUSÕES.....</b>	<b>79</b>
<b>6.1 Trabalhos futuros .....</b>	<b>80</b>
<b>REFERÊNCIAS .....</b>	<b>81</b>



## 1 INTRODUÇÃO

Todos os dias acontecem milhares de compras em lojas, supermercados ou através da *internet* em sites especializados em vendas *online*. Aparentemente uma operação simples, entretanto poucas pessoas têm a ciência de todas as operações e complexidades para que uma mercadoria chegue até o consumidor final.

Devido à imensa extensão territorial de nosso país, combinado com a baixa exploração de formas mais eficazes para distribuição de nossa produção, a maneira mais utilizada para a movimentação de mercadorias é o transporte rodoviário. A utilização de Centros de Distribuição (CD) torna-se vital para disponibilizar as mercadorias de maneira mais rápida ao cliente.

Os CDs são estruturas para recebimento e estocagem de mercadorias de diferentes fornecedores para a redistribuição (HUGOS, 2006). Este tipo de instalação viabiliza armazenar mercadorias em maior escala e variedade, possibilitando a distribuição para outros centros, varejo ou até ao consumidor final.

Um CD possui cinco atividades fundamentais (BARROS, 2005): recebimento, etapa na qual a mercadoria é conferida em sua chegada; movimentação, este processo é responsável pelo descarregamento e endereçamento para a armazenagem; armazenagem, é a estocagem no local adequado; seleção de pedidos, responsável pela retirada das mercadorias do estoque para compor pedidos; expedição, nesta fase os pedidos são conferidos e despachados para o transporte.

As atividades de recebimento, movimentação e estocagem têm a finalidade de garantir que a mercadoria seja armazenada no centro de distribuição. Estas atividades estão diretamente ligadas a problemas relacionados a estocagem onde, conforme Queiroz et al. (2009), um dos objetivos é maximizar a ocupação de espaço.

### 1.1 Motivação

Um sistema logístico visa agregar valor para as mercadorias, entretanto somente o processo de estocagem em si não é capaz de acrescentar valor, principalmente quando a mercadoria armazenada é a mesma que será expedida, ou seja, não sofre nenhuma transformação, onde muitas, vezes tende a diminuir devido a quebras, riscos de obsolescência, deterioração, entre outros motivos. No entanto, o fato de disponibilizar a mercadoria conforme

a solicitação do cliente, no prazo e local combinados, através das atividades de armazenagem, seleção de pedidos e transporte permite cumprir a proposta de valorização através do serviço de distribuição (CARVALHO et al., 2012).

Para uma empresa que tem como principal serviço a distribuição de mercadorias, possuir um CD que proporcione que seus processos sejam realizados no menor tempo e custo possível se torna uma busca constante, com a finalidade de manter a sua competitividade.

As empresas atacadistas têm como principal operação a logística de distribuição, onde constantemente são forçadas a rever seus processos para se manterem competitivas no mercado. A gestão de custos se torna primordial para manter o valor das mercadorias oferecidas competitivo e atraente para os clientes. Dentre os custos operacionais, pode-se destacar a atividade de seleção dos pedidos, que em muitos casos representam mais da metade dos custos totais de um armazém (KOSTER et al., 2007).

## **1.2 Problema**

Apesar das possibilidades de automatização, estudos apontam que aproximadamente 80% de todas as coletas de pedidos são realizadas de forma manual, através dos colaboradores que caminham ou se deslocam pelo depósito utilizando um carrinho a fim de recuperar os itens das prateleiras (NAPOLITANO, 2012). Por este motivo, a seleção de pedidos é uma operação intensa que envolve muitos colaboradores, sendo assim o processo mais caro dentro do depósito (RICHARDS, 2011).

É possível tornar a seleção de pedidos mais eficiente com políticas operacionais apropriadas que são classificadas em quatro grupos: *layout*, armazenamento, definição de rotas e processamento em lotes. O tempo necessário para separar um pedido pode ser dividido em três fases: o tempo para se deslocar em procura dos itens, o tempo para recolher os itens e as demais atividades. Deste modo existe uma oportunidade de melhora no processo, reduzindo as distâncias percorridas pelos colaboradores (DUKIC; CESNIK; OPETUK, 2010). Fatores humanos, como ergonomia, podem ter impactos significativos no desempenho da operação já que a falta de cuidados pode ocasionar lesões ou afastamento dos colaboradores por distúrbios musculares ou posturais (NEUMANN; DUL, 2010).

Outro fator decisivo e importante para conseguir, dentro das empresas, melhorias significativas no processo de seleção de pedidos é a utilização da Tecnologia da Informação



(TI). A TI é primordial para o controle das transações, suporte a gestão, apoio a tomadas de decisões e melhora no fluxo da comunicação (BOWERSOX et al., 2010).

Neste cenário surge a questão de pesquisa: “Como as metaheurísticas podem ser aplicadas em um protótipo computacional abordando o problema do *layout* de armazém para minimizar os custos no processo de seleção de pedidos? ”

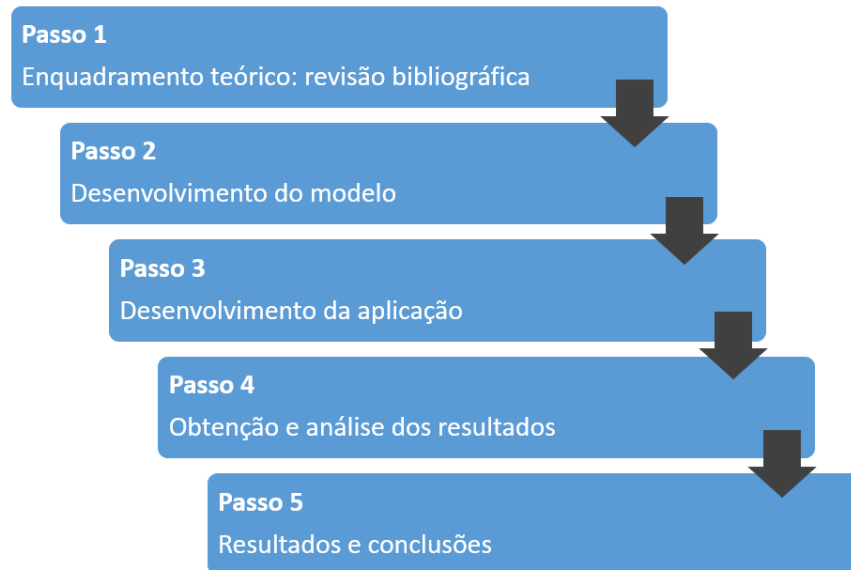
### **1.3 Objetivos**

O presente estudo tem a finalidade de criar uma aplicação computacional empregando metaheurística, capaz de propor a distribuição de mercadorias dentro de um CD com a finalidade de otimizar o processo de seleção de pedidos.

### **1.4 Método de Trabalho**

De maneira a atingir os objetivos, a Figura 1 apresenta as etapas realizadas para o desenvolvimento deste trabalho. O primeiro passo foi a revisão bibliográfica, onde foram revisados conceitos sobre o domínio do problema, que é a logística especificamente de um CD, seguido das técnicas e conceitos para abordar o problema e finalizando com as metaheurísticas.

O segundo passo foi o desenvolvimento do modelo para a realização da aplicação. O terceiro passo foi o desenvolvimento da aplicação, que é a transformação do modelo em uma aplicação computacional através de uma linguagem de programação. O quarto passo foi a obtenção dos resultados após realização dos testes. A última etapa foi a geração dos resultados e conclusões.

**Figura 1: Metodologia adotada na presente dissertação**

**Fonte:** Elaborada pelo autor.

O estudo das metaheurísticas, especificamente os Algoritmos Genéticos, é de suma importância para a realização do objetivo do trabalho, onde através deles poderão ser geradas soluções para o *layout* do CD, a partir de uma lista de mercadorias e as dimensões do armazém.

## 1.5 Organização do texto

Esta dissertação está dividida em seis capítulos. No segundo capítulo é abordado o assunto logística, onde é realizado um breve estudo sobre a sua evolução até chegar ao Centro de Distribuição.

No terceiro capítulo são estudados as técnicas e conceitos para abordar o problema, onde são encontrados os tópicos Tecnologia de Grupo devido a sua relação com manufatura que se faz presente neste trabalho toda vez que realiza-se inferência à seleção de pedidos, e o outro tópico é o Problema do Empacotamento que está ligado diretamente com a armazenagem das mercadorias, ou seja, com o Centro de Distribuição.

Outro tópico encontrado ainda no terceiro capítulo são os Algoritmos Genéticos. O emprego das metaheurísticas se faz necessário devido à complexidade combinatória do problema abordado, onde um Centro de Distribuição pode possuir milhares de mercadorias diferentes. As possíveis combinações de armazenagem inviabilizam desenvolver uma aplicação

capaz de gerar uma boa solução em um tempo viável o que credencia a utilização das metaheurísticas para o trabalho proposto.

O modelo de solução é apresentado no quarto capítulo, trazendo os detalhes de como o protótipo que foi desenvolvido, trazendo as suas funcionalidades e limitações. Também no mesmo capítulo são apresentados os aspectos inerentes ao desenvolvimento da aplicação como engenharia de *software*; arquitetura da aplicação, linguagem de programação entre outros.

O quinto capítulo apresenta os resultados e testes realizados de acordo com o modelo estabelecido. Nos resultados são apresentadas as instâncias utilizadas para os testes bem como os detalhes de calibração e de todos os experimentos realizados. Por fim, o último capítulo apresenta as conclusões e os trabalhos futuros.



## 2 LOGÍSTICA

Neste capítulo serão apresentados os conceitos e evolução da logística até chegar no conceito de um CD e sua relevância para a distribuição das mercadorias, desde a sua fabricação até a entrega para o consumidor final.

### 2.1 Conceito e Evolução da Logística

No passado, as limitações geográficas e climáticas dificultavam a produção de determinadas mercadorias. Os meios de transportes precários faziam com que as mercadorias produzidas fossem consumidas na sua volta sem que pudessem ser comercializadas a outras regiões, forçando as pessoas viverem perto das fontes produtoras. Apesar das dificuldades, a logística era um fator importante para o comércio de bens e serviços, entretanto o termo “logística” ainda não era utilizado (BALLOU, 2006).

Segundo Paoleschi (2009), a logística tem a missão de dispor a mercadoria ou serviço, no local exato, no tempo estabelecido e na forma acordada, contribuindo de forma efetiva nas operações da empresa.

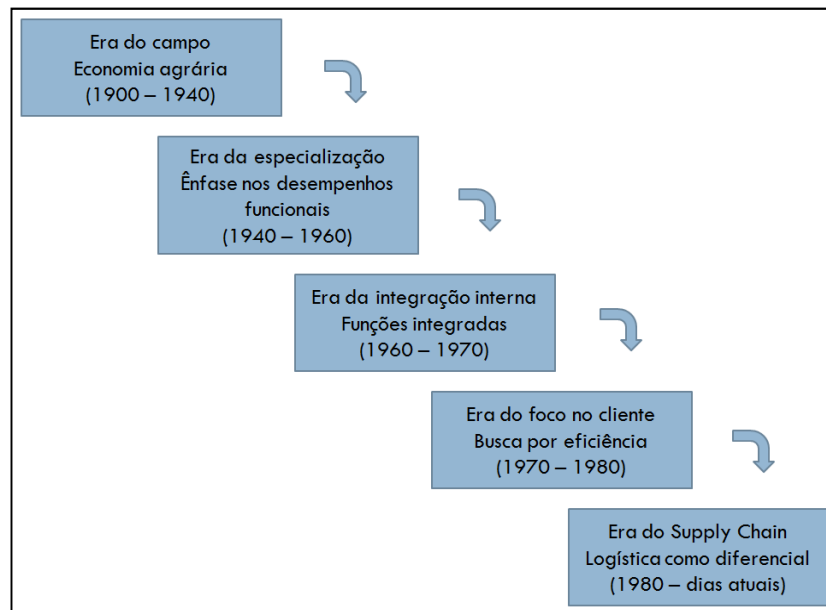
Ching (2007) reforça que a logística possui a atribuição de toda a movimentação de materiais, seja ela em ambiente interno ou externo da empresa, iniciando no recebimento da matéria-prima até o envio da mercadoria para o cliente.

No Brasil, a logística teve a sua expansão de forma significativa com a difusão da informatização, no início dos anos 80, a partir daí surgem as principais entidades: Associação Brasileira de Supermercados (ASBRAS), Associação Brasileira de Logística (ABRALOG), Instituto de Movimentação e Armazenagem (IMAM), dentre outras, todas com o trabalho de contribuir com a disseminação deste conceito, com foco nas organizações (SOUZA, 2012).

Na década de 80, apenas com o foco nas metodologias e modais de transportar, e armazenar. Na década de 90, começaram a se fazer cálculos, pois daí iniciou o conhecimento científico, estudos das relações, dispersões, movimentos etc., com foco em administração de matérias, distribuição, movimentação e armazenagem de matérias. Hoje muito mais complexo e amplo, com foco em controle, planejamento, tecnologia da informação, finanças e serviço ao cliente. Todas essas evoluções, aliadas ao processo de globalização, trouxeram novos desafios para as organizações, que é a competitividade no mercado globalizado (SANTOS, 2010).

A Figura 2 apresenta a evolução do processo logístico, que no início do século passado tinha a responsabilidade de distribuir as produções agrícolas, onde a partir da década de 1940 começou-se a buscar melhor desempenho na distribuição passando por inúmeras fases até chegar na logística que se conhece atualmente.

**Figura 2: Evolução do pensamento logístico**



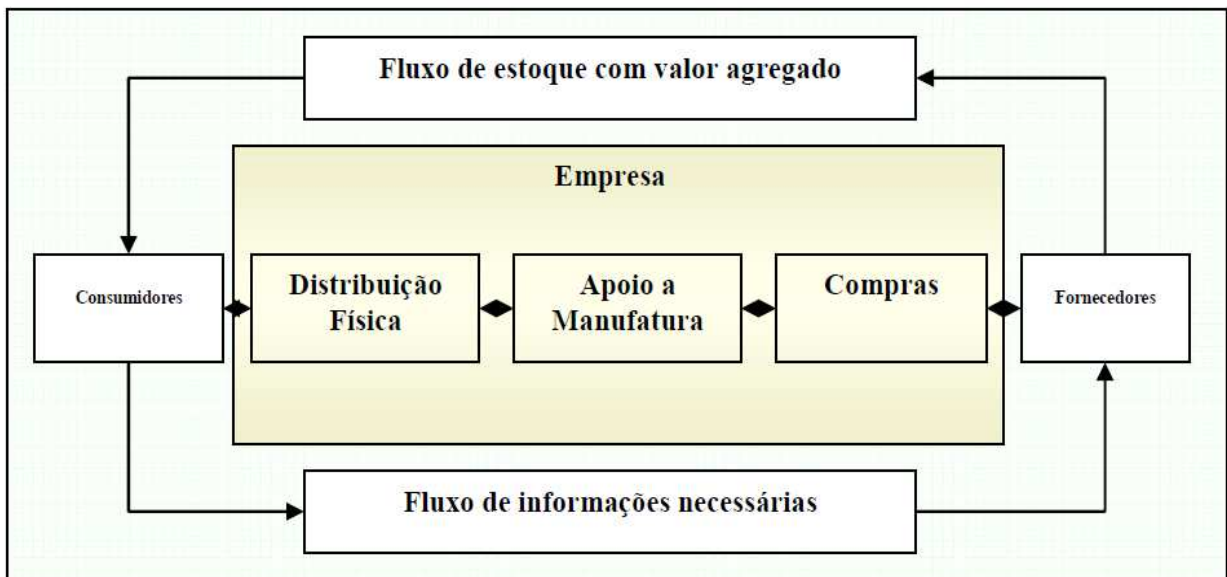
**Fonte:** Adaptado de Figueiredo e Arkader (1998).

A logística militar, desenvolvida na Segunda Guerra Mundial, causou maior visibilidade para a logística, iniciando muitos conceitos utilizados atualmente. Entretanto sua utilização em empresas comerciais se deu somente anos mais tarde. Nos meados de 1950, uma nova filosofia emergia "o conceito de *marketing*", onde a logística passa a ser oferecida como serviço. Entre os anos de 1950 e 1970 houve um grande avanço na logística, pois foi um período de enormes mudanças, ocorrendo transformações entre a teoria e a prática (BALLOU, 1995; LAMBERT; STOCK; VANTINE, 1998).

Conforme Fleury et al. (2000), a logística integrada surgiu no início da década de 1980, tendo rápida ascensão, em virtude das novas exigências de melhorias de processos de distribuição alavancadas pela tecnologia da informação. Segundo Boewrsox e Closs (2001), a logística integrada é a capacidade de vincular a empresa a seus fornecedores e clientes, conforme a Figura 3.

A Figura 3 apresenta o sistema que é utilizado pelas empresas para a distribuição de suas mercadorias, onde fica bem claro que os fornecedores são responsáveis por abastecer o mercado com suas mercadorias, e em contrapartida, os clientes fornecem informações quanto às mercadorias consumidas e no centro deste fluxo está a empresa responsável pela distribuição.

**Figura 3: Sistema Logístico**



**Fonte:** (BOWERSOX; CLOSS, 2001).

A gestão empresarial vem sofrendo constantes transformações que envolvem todas as áreas, as teorias e técnicas de administração das empresas. A logística cada vez mais vem se tornando fundamental para este processo. O Mercado vem se tornando mais competitivo forçando as empresas a estarem constantemente revendo seus processos a fim de minimizarem custos e possuírem processos mais eficientes.

## 2.2 Logística de Distribuição

Devido à concorrência acirrada entre as empresas, a fidelização de clientes ou a ampliação dos clientes atendidos se faz necessária, onde ser competitivo se torna vital e o diferencial passa a ser o valor agregado a serviços e/ou mercadorias acabadas. Neste cenário, a logística de distribuição passa a ser fundamental, seja na ampliação ou na manutenção de clientes (KAMINSKI, 2004).

Para Ching (2007), o principal foco da logística de distribuição é tratar as relações entre empresa e o seu cliente, possuindo a responsabilidade da distribuição das mercadorias até os pontos de venda ao consumidor, garantindo a entrega dos pedidos dentro dos prazos acordados e atendendo todas as solicitações dos clientes.

A distribuição física tem seu início na solicitação do cliente, que se concretiza em um pedido, que será processado pela empresa passando pelas etapas de separação, conferência, expedição e envio para o cliente. As operações refletem nas áreas de *marketing*, vendas e diretamente na precificação das mercadorias pelas disponibilidades, ou ainda quando será regularizado o seu estoque (BOWERSOX et al., 2006). A finalidade da logística de distribuição é entregar as mercadorias conforme as solicitações dos clientes, atendendo as suas expectativas quanto à qualidade das mercadorias e prazos acordados no menor custo possível.

### 2.3 Armazenagem

Devido à exigência de prazos curtos, impulsionados por canais alternativos de venda como telefone, catálogos e *internet*, e o aumento do número de mercadorias ofertadas houve uma alavancagem das demandas em armazenagem. Desta forma, com o aumento das exigências dos clientes, os armazéns de mercadorias acabados estão sendo gradativamente substituídos por centros de distribuição (LIMA, 2006).

A armazenagem vem evoluindo ao longo dos anos, passando de uma pequena participação para uma das funções mais importantes atualmente no sistema logístico (SILVA, 2010).

A importância da armazenagem se justifica em manter o fluxo logístico abastecido, mantendo a uniformidade e continuidade, atendendo as necessidades da cadeia e agregando valor à mercadoria (GASNIER; BANZATO, 2001).

Entretanto, para Ballou (2006) existem quatro motivos para justificar que a empresa destine o seu espaço físico para a armazenagem.

- a) Redução de custos de produção e transporte: redução dos custos de transporte devido aos gastos com produção e estocagem, conseqüentemente reduzindo os custos gerais na cadeia de distribuição;
- b) Gerenciamento de suprimento e demanda: devido à sazonalidade, há dificuldade para coordenar o suprimento mediante a demanda, onde níveis de estoque de segurança são necessários no processo de distribuição;



- c) Necessidades no processo produtivo: a armazenagem é empregada em processos produtivos. Entretanto a aplicação não se restringe somente a guardar mercadorias acabadas no final de um processo de fabricação, mas são utilizados em mercadorias taxadas, assegurando a comercialização;
- d) Considerações de *marketing*: é importante para o setor de *marketing* que a mercadoria esteja disponível. Fatores como escassez no mercado ou disponibilidade de entrega imediata podem agregar valor a mercadoria.

## 2.4 Layout de Armazém

O estudo do *layout* apresenta uma relação entre o tamanho do armazém e o tamanho das estruturas de armazenagem. Estas pesquisas auxiliam no dimensionamento dos armazéns e são utilizadas como base para outros modelos na literatura que têm como foco o *layout* de armazém.

Na fase de projeto, uma decisão importante a ser tomada é com relação à proporção entre a largura e a profundidade. Devido a fator de deslocamento, que irá impactar diretamente na produtividade, Koster (2010) afirma que um armazém deve ter uma proporção de 2 para a largura total do armazém (L) e 1 para profundidade total do armazém (B), ou seja,  $L = 2B$ .

Além do fator da proporção, Koster (2010) enfatiza ainda que a proporção entre a largura do conjunto de armazenagem (*l*) e as dimensões entre dois conjuntos separado pelo corredor de *picking* (*g*) afeta como serão realizadas as dimensões do armazém, como é apresentada na equação abaixo.

$$L = \sqrt{\frac{Plb}{n}} \quad e \quad B = \frac{1}{2} * \sqrt{\frac{Plb}{n}} \quad (2.1)$$

onde:

- *P*: número de posições de armazenagem necessárias;
- *n*: Número de níveis do armazém.

Uma relação do terreno e da área de armazenagem, incluindo as áreas de carga e descarga são consideradas tal que:

$$B_{1\geq} b_1 + NN * a_2 + S \quad (2.2)$$

onde:

- $B_1$ : Largura do edifício de armazenagem;
- $b_1$ : Largura do corredor principal de passagem;
- NN: Número de boxes de armazenagem;
- $a_2$ : Largura do Box de armazenagem;
- S: Largura da área administrativa.

Considerando os limites do terreno:

$$B_1 \geq b_1 + 2 * c + b_2 \quad (2.3)$$

onde:

- c: largura do corredor de circulação interna.

## 2.5 Centro de Distribuição

De acordo com Hugos (2006), centro de distribuição são instalações com a finalidade de receber mercadorias de diferentes fornecedores. Em um CD, são recebidas grandes quantidades de mercadorias onde acabam reduzindo expressivamente os custos com transporte considerando o fato que na maioria das vezes os fornecedores estão situados a uma longa distância do consumidor final.

Segundo Barros (2005), um CD viabiliza gerir os estoques das mercadorias acondicionadas, gerando novas frações e organizando-as para direcioná-las a pontos de vendas mais próximos. O ponto de venda pode ser uma filial da própria empresa, uma loja especializada em vendas da mercadoria ou até mesmo diretamente ao consumidor final.

Segundo Sellitto et al. (2009), um CD tem como principais objetivos: maximizar o uso dos equipamentos internos; diminuir o espaço necessário; difundir a utilização da informação; diminuir o tempo necessário entre a solicitação e carregamento dos pedidos.

As instalações de armazenagem desempenham papel primordial no atendimento, de forma eficiente e eficaz, aos desejos desse mercado cada vez mais competitivo. Logo, se torna primordial para as empresas que atuam na cadeia de distribuição possuir um CD tornando assim possível equilibrar os custos da operação com o nível de serviço desejado.

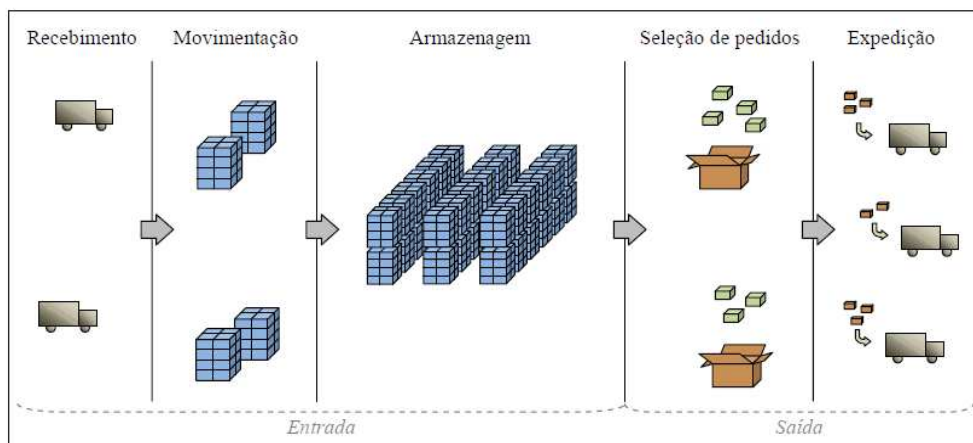
Ao encontro às citações dos parágrafos anteriores, um CD se torna primordial na cadeia de distribuição de um determinado fornecedor até o seu consumidor final. Dentro de um CD são realizadas atividades que vão desde a gestão da chegada da mercadoria até a saída através dos pedidos oriundos dos clientes.

Conforme relata Barros (2005), um CD possui basicamente cinco operações principais, sendo elas: recebimento, movimentação, armazenagem, separação de pedidos e expedição. O recebimento é responsável pela conferência da mercadoria referida na nota fiscal no momento da chegada à empresa. A movimentação é a etapa em que a mercadoria é transferida para a armazenagem em locais previamente demarcados. A armazenagem tem a responsabilidade de transferir a mercadorias das áreas de movimentações para as destinadas a separação de pedidos. Quanto à separação de pedidos, trata-se da operação com a finalidade de retirar do estoque as mercadorias para compor as requisições dos clientes. A expedição é o último processo dentro de um CD, onde os pedidos são conferidos e despachados para o transporte até o cliente final.

A Figura 4 (adaptado de WURMAN; D'ANDREA; MOUNTZ, 2008) mostra as atividades básicas dentro de um CD tradicional, onde as mercadorias são recebidas, movimentadas e armazenadas dentro do processo de entrada, gerando assim mercadorias disponíveis para a venda. Já a etapa de saída se dá a partir da seleção dos pedidos que são direcionados para a expedição para conferência e despacho dos pedidos.

**Fonte:** adaptada de Wurman et al (2008).

**Figura 4: Fluxo de mercadorias em um centro de distribuição tradicional**



## 2.6 Seleção de pedidos

Dukic et al. (2010) e Koster, Le-Duc e Roodbergen (2007) afirmam que o processo de seleção de pedidos é definido como o processo de recolhimento de mercadorias, a partir de locais de armazenagem, em resposta a um pedido específico do cliente. De acordo com Pocinho (2013), é responsabilidade do armazém garantir os níveis de estoque para atender os pedidos dos clientes no menor tempo possível.

A seleção de pedidos na maioria dos armazéns é considerada o processo que exige mais mão-de-obra e maior investimento de capital, os custos relacionados a esta operação estão estimados em torno de 55% do custo total do CD. Portanto, é prioridade as melhorias no processo de seleção de pedidos, bem como a sua eficiência, e para tal, é necessário reduzir o tempo necessário para a realização do processo (POCINHO, 2013).

Escolher um sistema de seleção de pedidos apropriado é vital para manter a eficiência do armazém. A implementação de um sistema de seleção de pedidos visa maximizar o nível de serviço ao cliente, minimizar as distâncias percorridas e minimizar os custos.

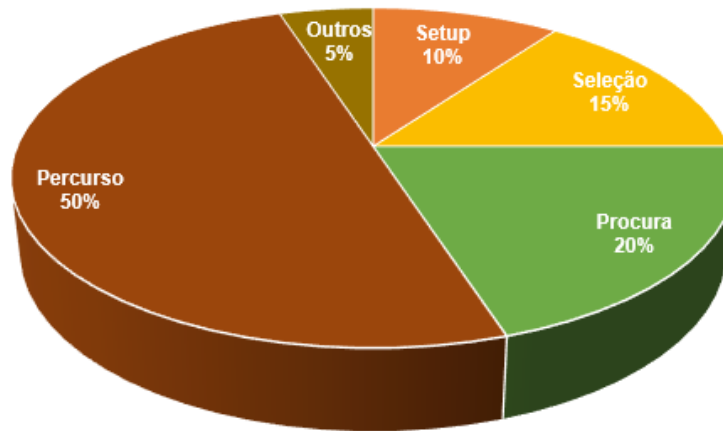
A automatização do processo de seleção de pedidos é frequentemente analisada como forma de reduzir os custos de mão-de-obra. Muitas empresas continuam a separar os seus pedidos manualmente, devido à variabilidade da forma e tamanho das mercadorias, à sazonalidade ou devido ao grande investimento para automatizar.

A seleção de pedidos automatizada é realizada através de sistemas controlados por computadores para recolher as mercadorias, onde este processo ocorre de forma totalmente automática. Por outro lado, a forma manual é realizada através dos colaboradores que caminham ou se deslocam pelo depósito utilizando um carrinho a fim de recuperar os itens das prateleiras (NAPOLITANO, 2012).

Tompkins et al. (2003) enfatizam que o tempo total de processamento de um pedido é o tempo necessário que o colaborador leva para realizar o trajeto de coleta das mercadorias. O tempo total de coleta das mercadorias está dividido da seguinte forma:

- a) Percurso: tempo que o colaborador está em deslocamento para o recolhimento das mercadorias;
- b) Procura: tempo necessário para identificar as mercadorias;
- c) Seleção: tempo para a coleta das mercadorias;
- d) *Setup*: o tempo para as tarefas administrativas e de preparação;
- e) Outros: imprevistos, atrasos, etc.

**Figura 5: Percentual dos tempos típicos no processo de seleção de pedidos**



**Fonte:** Elaborada pelo autor.

A Figura 5 apresenta a distribuição dos tempos para a realização do processo de seleção de pedidos conforme Tompkins et al. (2003). É possível verificar que o fator de maior impacto é o tempo que o colaborador fica em deslocamento. Neste ponto, é possível verificar a importância do *layout* do armazém, ou seja, da localização das mercadorias para o processo.

## ***2.7 Warehouse Management System***

Para Ribeiro, Silva e Benvenuto (2006), em razão da política de estoque mínimo, os clientes fazem mais pedidos e em menores quantidades obrigando a cadeia imediata, ou seja, o atacadista, a possuir estoques para suprir as suas demandas.

O aumento da frequência dos pedidos e a variação das embalagens comercializadas no varejo aumenta o número de mercadorias a serem controladas, as operações em um Centro de Distribuição, impactando diretamente em maiores custos, redução da produtividade e espaços cada vez maiores (SILVA, 2010).

Na atualidade, existem diversas soluções para auxiliar e automatizar o gerenciamento de um Centro de Distribuição. Dentre todas as soluções disponíveis no mercado a que mais se destaca é o *Warehouse Management System (WMS)*. Uma definição para este tipo de solução é referida por Barros (2005).

O *WMS* é um sistema de gestão de armazéns, que otimiza todas as atividades operacionais (fluxo de materiais) e administrativas (fluxo de informações) dentro do processo de armazenagem, incluindo atividades como: recebimento, inspeção, endereçamento, armazenagem, separação, embalagem, carregamento, expedição, emissão de documentos, controle de inventário (BARROS, 2005).

Um *WMS* é um sistema que viabiliza a otimização da gestão de um Centro de Distribuição padronizando as operações e os fluxos de informações. Porém, um sistema de *WMS* não necessariamente possui automatização de processos físicos.

É possível também definir um *WMS* como um conjunto de *softwares* interligados com o intuito de gerenciar as informações, que controlam eletronicamente as operações de um Centro de Distribuição, pois aplica modernas ferramentas gerenciais, que fazem a distribuição das tarefas de forma eficiente, reduzindo o uso de colaboradores no processo, minimizando falhas e agilizando processos.

Segundo Ribeiro, Silva e Benvenuto (2006), as principais funcionalidades ofertadas por um *WMS* são:

- a) Rastreabilidade das operações: Todas as tarefas executadas são identificadas com os colaboradores ou equipamentos que as realizaram, em tempo real;
- b) Inventários: utilizando regras definidas pela empresa, o sistema faz convocações para inventários parciais ou gerais, sendo eles orientados a itens ou endereços;
- c) Planejamento e controle das capacidades: Com base nos cadastros dos locais de recebimento, expedição, colaboradores, empilhadeiras, etc., e também da definição do consumo dos recursos e tarefas, pode-se fazer um planejamento das tarefas;
- d) Definição dos locais de armazenagem: Através do mapeamento dos endereços de armazenagem e suas características auxiliares, a fim de gerenciar a distribuição das mercadorias;
- e) Classificação de itens: Permite a caracterização de mercadorias através de parâmetros que definam a classe à qual pertence facilitando a parametrização;
- f) Controle de lotes, quarentenas e qualidade: Manter registros dos produtos armazenados quanto à fabricação, ou seja, os lotes auxiliam na movimentação de mercadorias com problemas para a área de quarentena já que as informações armazenadas auxiliam na identificação.

### 3 TÉCNICAS E CONCEITOS PARA ABORDAR O PROBLEMA

Este capítulo tem a finalidade de apresentar as técnicas e conceitos que foram necessários para auxiliar na geração do modelo. Dentre os conceitos que foram abordados, estão a Tecnologia de Grupo, Problema de *Layout* de Armazém e Trabalhos relacionados.

#### 3.1 Tecnologia de Grupo

O conceito de Tecnologia de Grupo (TG) teve suas primeiras aplicações, no início do século passado, a partir do ano de 1920. Em 1925, nos Estados Unidos, Flanders apresentou um trabalho que aplicava o método na organização de manufatura à Sociedade Americana de Engenharia Mecânica. No ano de 1937 na União Soviética, A. Sokolovskiy definiu as principais características da TG, sugerindo que sequências padronizadas de operações poderiam produzir partes similares, contribuindo assim para que as técnicas de *flow shop* pudessem ser desenvolvidas em indústrias que processavam lotes pequenos. Já na França, precisamente em Paris no ano de 1949, o sueco A. Korling apresentou um trabalho de “grupos de produção”, o qual adaptava técnicas de linha de produção para manufatura em lotes (GROOVER, 2012).

A TG pode ser definida como uma filosofia para a resolução de problemas que têm como princípio a descoberta das semelhanças visando à otimização do uso de processos operacionais mediante a um tratamento de grupo (DALMAS, 2004). Levando em consideração o ponto de vista da manufatura, a TG é uma metodologia que permite gerir as tarefas de um sistema produtivo, em que as similaridades das peças permitem gerar pequenos lotes, onde ao serem direcionadas para as células de produção permitem o aumento da produção pelo fato de necessitarem de operações similares gerando assim vantagens econômicas para a empresa.

É possível encontrar inúmeras definições para a TG, onde para alguns autores o fato de agrupar equipamentos não necessariamente define uma célula. De forma geral, é possível definir com maior abrangência que uma célula é composta por equipamentos distintos, reunidos conforme as necessidades do fluxo de produção no processamento das mercadorias conforme suas similaridades (BRAGHIROLI, 2009).

Conforme Groover (2012), a TG oferece inúmeros benefícios aos processos produtivos das empresas que optarem em implantá-la, onde são destacados:

- a) Padroniza as ferramentas, sistemas de fixação e o *setup*;

- b) Redução no manuseio dos materiais devido às dimensões das células serem menores que a fábrica inteira;
- c) Planejamento de processos e cronograma de produção é simplificado;
- d) Redução no tempo do *setup*, conseqüentemente aumento do tempo produtivo;
- e) Redução dos estoques intermediários;
- f) A satisfação dos colaboradores aumenta ao colaborarem em uma célula TG;
- g) Aumento na qualidade do trabalho.

Ao se implantar a TG, existem duas iniciativas importantes que serão obstáculos para esta implantação, a primeira é a identificação das famílias de partes que irão consumir bastante tempo do processo e a segunda será a reorganização do *layout* que deixará os equipamentos indisponíveis durante o processo de realocação (GROOVER, 2012).

### 3.1.1 Família de Partes

De acordo com Semaan (2010), o problema de formação de *clusters* é do tipo NP-Completo devido ao grande número de combinações dado um número de elementos a serem classificados. Em seu trabalho, Dias (2004) apresenta dois exemplos, no primeiro o número de clusters está predefinido como dois e dez é o número de elementos a serem classificados, neste cenário o número de combinações possíveis é 511. No segundo exemplo, com os mesmos 10 elementos, entretanto sem predefinição de número de clusters, o número de possíveis combinações sobe para 115.975.

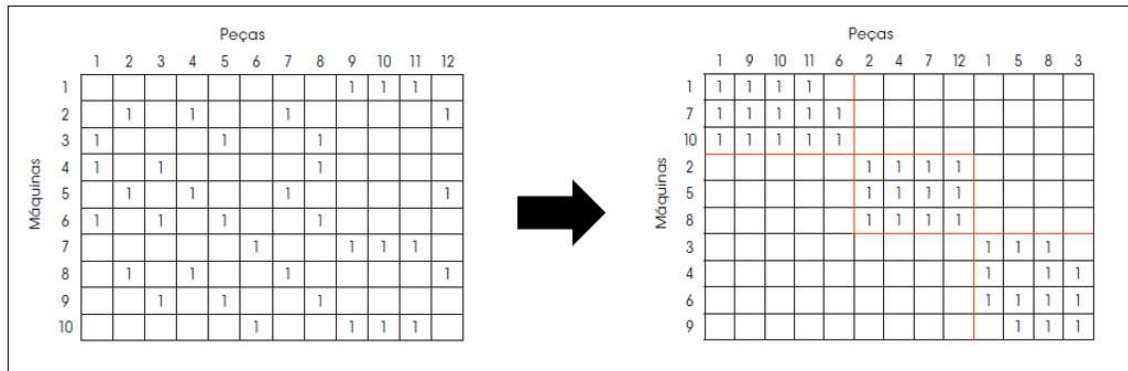
O agrupamento de peças ou partes conforme similaridade pode ser definido como uma Família de Partes (FP). As partes que compõem uma FP não necessitam ser iguais, entretanto, devem possuir aspectos similares os quais justifiquem pertencerem ao mesmo grupo, como restrições físicas referentes ao processo de fabricação. Em um ambiente de manufatura, normalmente os critérios aplicados para a formação das FP são a forma geométrica e o processo produtivo (BORGES, 2009).

Devido a inúmeras metodologias utilizadas, a formação de FP podem apresentar resultados diferentes, mesmo usando uma mesma base de dados. Os resultados apresentados estarão diretamente ligados à combinação das metodologias associadas a diferentes critérios e limitações, resultando em um maior grau de processamento ou complexidade de análise decisória. Os algoritmos se tornam ferramentas fundamentais na definição das FP e células, permitindo se valer de todos os benefícios do uso da Tecnologia de Grupo (BRITO, 2010).



Os primeiros algoritmos têm como base principal a manipulação de matriz, onde as linhas e colunas são alteradas a fim de resultar uma diagonal de blocos onde o resultado é a família de partes. A Figura 6 apresenta uma simulação da matriz antes e depois de agrupada.

**Figura 6: Matriz de incidência antes e depois do agrupamento**



**Fonte:** Elaborada pelo autor.

Nas últimas décadas, inúmeras soluções visando resolver o problema da criação das Famílias de Partes, consideram inúmeros critérios de desempenho como custo da movimentação, utilização das máquinas, tempo de *setup* entre outros. Chu (1989) define esses métodos em cinco tipos: os que manipulam matrizes, agrupamento hierárquico e não hierárquico, programação matemática, técnicas gráficas e heurísticas.

O estudo das FP se fez necessário neste trabalho para a classificação das mercadorias em grupos a fim de gerarmos o armazenamento respeitando as características de cada grupo. Ter realizado uma revisão literária em algoritmos capazes de realizar as FP foi primordial para o desenvolvimento do protótipo.

### 3.2 Bin Packing

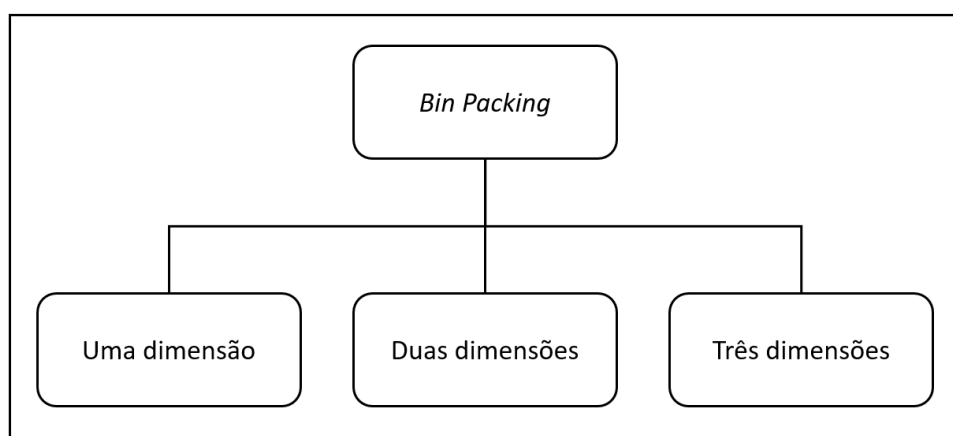
No *Bin Packing*, objetos que possuem diferentes volumes devem ser acondicionados em um número finito de embalagens, onde o objetivo é minimizar esta operação, ou seja, reduzir o número de embalagens necessárias.

O *Bin Packing* pode ser considerado um problema de tomada de decisão ou de otimização. No caso do problema quanto a tomada de decisão, uma determinação deverá ser analisada, se todos os objetos irão ser comportados pelo recipiente, ou seja, em outras palavras o problema verifica se o recipiente possui espaço suficiente para acomodar o objeto em questão

onde a resposta será booleana. Por outro lado, o problema de otimização visa minimizar a quantidade de recipientes ou o espaço desperdiçado dentro dos recipientes. Matemáticos e cientistas da computação enquadram o *Bin Packing* como um problema de otimização (GEHRING; BORTFELDT, 1997).

De acordo com Cunha et al. (2008), o problema de empacotamento tem muita aplicabilidade no campo da logística de distribuição, quando se tem o objetivo de otimizar a frota necessária para a realização das entregas. O *Bin Packing* geralmente é classificado de acordo com o número de dimensões conforme é possível visualizar na Figura 7.

**Figura 7: Classificação do *Bin Packing* de acordo com as dimensões**



**Fonte:** Elaborada pelo autor.

### 3.2.1 Uma dimensão

O problema de uma dimensão otimiza somente um parâmetro por vez, por exemplo, ao gravar arquivos em uma unidade de armazenamento onde o único parâmetro a ser considerado é o tamanho de cada arquivo, onde ao selecionar corretamente os arquivos podemos ter dois objetivos: gravar o maior número possível de arquivos e ou deixar o menor espaço possível sem utilizar. O problema de uma dimensão irá otimizar o comprimento ou a largura do recipiente.

### 3.2.2 Duas dimensões

O problema com duas dimensões trata-se da adição de mais uma variável no problema para ser otimizado. Na versão com duas dimensões, ambos os parâmetros são considerados onde precisam ser otimizados simultaneamente a fim de alcançar uma boa solução. Um

exemplo para contextualizar podemos considerar um recipiente plano como, por exemplo, uma mesa retangular, onde existem inúmeros recortes de papelão menores retangulares com dimensões variadas, onde o objetivo é conseguir ocupar a maior área. Um outro exemplo é clássico game “Tetris”, no entanto a embalagem exige que cada objeto seja ortogonal, de modo que os objetos podem ou não ser girados.

### 3.2.3 Três dimensões

No *Bin-Packing* tridimensional, como seria de se esperar, cada objeto e cada recipiente possuem três dimensões. Esses objetos e o recipiente são representados com no mínimo três atributos que armazenam os valores de cada dimensão sendo eles: largura, comprimento e altura. Cada objeto deve caber no recipiente de forma eficiente onde deve-se tentar ocupar o maior espaço possível sem exceder a capacidade do recipiente (BISCHOFF, 2004). Os problemas com três dimensões podem ser classificados da seguinte forma:

- a) Com base no número de recipientes utilizado
  - Problema de carregamento de um único recipiente;
  - Problemas de carregamento de múltiplos recipientes.
  
- b) Com base no tipo de objetos a serem embalados
  - Problemas homogêneos;
  - Problemas heterogêneos.

Cada um dos tipos de problemas possui suas características, complexidades e aplicabilidades. No primeiro caso, quanto ao recipiente, é possível exemplificar a operação portuária onde os produtos têm que ser colocados em recipientes denominados containers, podendo estar previamente alocados, conforme contrato, em um ou mais *containers*.

Já no segundo caso, é possível utilizar como exemplo um armazém que tem que realizar o empacotamento dos produtos para serem entregues a seus clientes onde podem ser consideradas duas hipóteses: que todos os produtos possuem dimensões similares ou existe diferença entre cada produto. Independentemente do caso, pode ser considerado que se possuem recursos finitos, ou seja, número de recipientes definidos ou recursos infinitos, ou seja, será possível usar quantos recipientes forem necessários.

Existem vários algoritmos e metodologias disponíveis para resolver os problemas de empacotamento tridimensional.

### 3.3 Problema de *Layout* de Armazém

Conforme Paletta e Silva (2003), *layout* é a disposição de recursos da produção com a sua interação em um espaço delimitado. O *layout* do armazém é de extrema importância para a eficiência logística pelo fato de estar diretamente ligado ao fluxo das mercadorias, proporcionando uma redução nos custos das operações de armazenagem e seleção de pedidos (MOURA, 2002).

Para se obter um posicionamento adequado das mercadorias, a fim de reduzir o deslocamento dentro do armazém, é necessário priorizar algumas mercadorias armazenadas, em detrimento de outras. A redução de investimentos em sistemas computacionais complexos através de modelos simplificados, muitas vezes contribui para não se obter ótimos resultados (OLIVEIRA; PALLETA, 2010).

Na gestão do CD, os colaboradores responsáveis pela operação devem decidir onde armazenar as mercadorias. A melhor localização é a que irá gerar o melhor custo benefício entre o acesso a mercadorias para a seleção de pedidos e o local para a armazenagem das mercadorias, a fim de minimizar os processos de armazenagem e seleção de pedidos. A este problema é dado o nome de Problema de *Layout* de Armazém (PLA). De acordo com Tompkins et al. (1996), o problema pode ser definido conforme o seguinte modelo matemático:

- a)  $q$  - número de locais de armazenagem;
- b)  $n$  - número de mercadorias;
- c)  $m$  - número de locais de entrada e saída ;
- d)  $S_j$  - número de locais de armazenagem da mercadoria  $j$ ;
- e)  $T_j$  - número de movimentações da mercadoria  $j$ ;
- f)  $p_i$  - entradas e saídas do armazém pelo ponto  $i$ ;
- g)  $d_{ik}$  - distância necessária a percorrer entre o ponto  $i$  e o local de armazenagem  $k$ ;
- h)  $x_{jk}$  - se a mercadoria  $j$  é atribuída ao local de armazenagem  $k=1$ , caso contrário 0;
- i)  $f(x)$  - distância média percorrida.

O PLA pode ser formulado conforme Tompkins et al. (1996), através da seguinte função:

$$\min = \sum_{j=1}^n \sum_{k=1}^q \frac{T_j}{S_j} \sum_{i=1}^m p_i d_{ik} \quad (3.1)$$

Sujeito a

$$\sum_{j=1}^n x_{jk} = 1 \quad k = 1, \dots, q \quad (3.2)$$

$$\sum_{k=1}^q x_{jk} = S_j \quad j = 1, \dots, q \quad (3.3)$$

$$x_{jk} \in (0,1) \quad \text{para todos os } j \text{ e } k \quad (3.4)$$

Considerando que cada mercadoria tenha a mesma probabilidade de se movimentar entre o ponto  $i$  e o local de armazenagem  $j$ , a probabilidade de um local de armazenagem afetar a mercadoria  $j$  a ser selecionada para a movimentação de saída e entrada é  $(1/S_j)$ . Assim a distância percorrida entre o local de armazenagem  $k$  e despacho é dada por:

$$f_k = \sum_{i=1}^m p_i d_{ik} \quad (3.5)$$

Para minimizar a distância média total percorrida, é necessário:

- Ordenar as mercadorias de acordo com o valor de  $T_j$  e de  $S_j$ .

$$\left(\frac{T_1}{S_1}\right) \geq \left(\frac{T_2}{S_2}\right) \geq \dots \geq \left(\frac{T_n}{S_n}\right) \quad (3.6)$$

- Calcular os valores de  $f_k$  para todos os locais de armazenagem.
- Atribuir a mercadoria 1 ao local de armazenagem  $S_1$  que tiver o menor valor de  $f_k$  e assim sucessivamente.

De acordo com Lai, Xue e Zhang (2002), trata-se de um problema da classe NP-Difícil pois se enquadra nos problemas clássicos de armazenagem, onde para cada mercadoria será disponibilizado de mais de um local de armazenagem, onde o problema não consiste somente em definir um local, mas também tratar a diversidade contida nas diferentes mercadorias a serem armazenadas, o que torna o problema ainda mais complexo.

### 3.4 Metaheurísticas

Conforme a etimologia (estudo da origem e evolução das palavras) a palavra Heurística vem do grego *heuriskein* que significa encontrar ou descobrir. É possível definir Heurística como processos ou métodos criados para encontrar a solução de problemas.

As metaheurísticas podem ser definidas como um método de solução que administra interações entre procedimentos de melhoria local e mecanismos para diversificar a exploração, a fim de escapar dos ótimos locais e realizar uma busca ampla no espaço de soluções (GLOVER; KOCHENBERGUER, 2003).

Segundo Linck (2012), as metaheurísticas têm a finalidade de solucionar problemas de difícil solução, explorando de forma geral um grande espaço de busca nessas instâncias. Esses algoritmos alcançam isto, realizando buscas eficientes reduzindo o espaço analisado.

De acordo com Brum (2014), a análise da complexidade dos algoritmos é abordada pelo ramo denominado Teoria da Complexidade Computacional, empregando considerações matemáticas. Esta teoria realiza a classificação dos problemas de otimização combinatória em classes de complexidades. Esta classificação é possível através da avaliação dos recursos computacionais necessários para solução, normalmente relacionados com o tamanho do conjunto de entrada. Esta classificação é conhecida como teoria da complexidade computacional, relacionando os problemas de Otimização Combinatória em quatro classes.

O termo Otimização Combinatória refere-se a uma área da matemática e da ciência da computação responsável por estudar problemas de otimização em conjuntos. Mesmo com a evolução dos computadores quanto ao poder de processamento, ainda existem problemas cujo tempo necessário para resolver não é aceito pelo usuário criando assim problemas intratáveis. De uma forma análoga, os problemas tratáveis têm o seu limite superior de complexidade na forma polinomial, enquanto os intratáveis são exponenciais (LINDEN, 2008).

Na Tabela 1, é apresentado por Talbi (2009) algumas relações entre complexidade, tamanho do conjunto de entrada e o correspondente tempo para solucionar. No pior caso apresentado que é a relação da complexidade  $O(3^x)$ , com o conjunto de entrada de 50 elementos o tempo para solução é de  $2 \times 10^8$  séculos, ou seja, o equivalente a 20.000.000.000 anos para encontrar uma solução.

Tabela 1: Tempo para a solução em relação ao conjunto de entrada.

Complexidade	Tamanho				
	10	20	30	40	50
$O(x)$	0,00001 s	0,00002 s	0,00003 s	0,00004 s	0,00005 s
$O(x^2)$	0,0001 s	0,0004 s	0,0009 s	0,0016 s	0,0025 s
$O(x^5)$	0,1 s	0,32 s	24,3 s	1,7 min	5,2 min
$O(2^x)$	0,001 s	1,0 s	17,9 min	12,7 dias	35,7 anos
$O(3^x)$	0,059 s	58,0 min	6,5 anos	3855 séculos	$2 \times 10^8$ séculos

Fonte: (TALBI, 2009).

Enfim, existe uma gama de problemas de otimização encontrados em diversas áreas como logística, telecomunicações, planejamento financeiro, produção, entre outros, que tem motivado o desenvolvimento de técnicas de otimização conhecidas como metaheurísticas.

Alguns exemplos de metaheurísticas são destacados por Brum (2014) e Linck (2012) como: Recozimento Simulado (*Simulated Annealing*), Colônia de Formigas (*Ant Colony Algorithm*), Algoritmo Genético (*Genetic Algorithm*), Busca Tabu (*Tabu Search*) e GRASP (*Greedy Randomized Adaptive Search Procedure*).

### 3.4.1 Algoritmos Genéticos

Os Algoritmos Genéticos (AG) permitem uma simplificação na modelagem e solução de problemas de otimização, devido a permitirem a representação de uma solução potencial para um determinado problema utilizando uma estrutura semelhante a um cromossomo, aplicando operadores de seleção e cruzamento de forma a preservar informações críticas relativas à solução dos problemas (FERNANDES, 2005).

Através de seu livro *Adaptation in Natural and Artificial Systems*, Holland (1975) estudou a evolução das espécies e apresentou um modelo heurístico com a finalidade de conseguir gerar boas soluções para problemas complexos. Não demorou muito para que o AG proposto transcendesse o papel para se tornar uma ferramenta muito utilizada pela computação (HOLLAND, 1975).

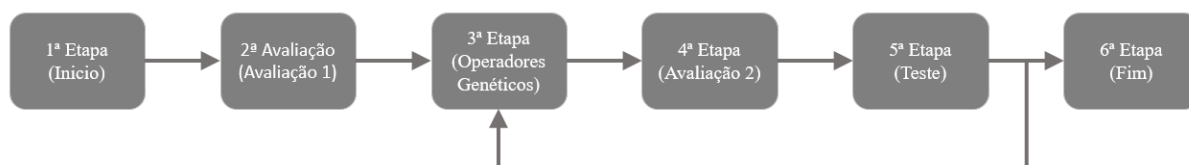
Conforme Linden (2008), os AGs são aplicados em problemas de otimização com muito sucesso desde a sua criação, entretanto, para o seu entendimento é necessário assimilar as denominações utilizadas pelo AG que têm origem do processo natural de evolução, conforme é apresentado na Tabela 2.

**Tabela 2: Análise do vocabulário do AG x problema computacional**

<b>Evolução Natural</b>	<b>Problema Computacional</b>
População	Conjunto de soluções
Indivíduo	Solução de um problema
Cromossomo	Representação de uma solução
Gene	Parte da representação de uma solução
Ambiente	Função objetivo ou aptidão ( <i>fitness</i> )
Cruzamento, Mutação	Operadores de busca

**Fonte:** Adaptada de Pacheco (1999).

O AG proposto por Holland (1975), de uma forma geral, pode ser representado conforme as etapas apresentadas na Figura 8.

**Figura 8: Etapas do AG proposto por Holland**

**Fonte:** Elaborada pelo autor.

- 1ª Etapa (Início): É gerada uma população inicial de forma aleatória onde os indivíduos gerados são possíveis soluções para o problema;
- 2ª Etapa (Avaliação): São avaliados todos os indivíduos da população através da função *fitness*;
- 3ª Etapa (Operadores Genéticos): Nesta etapa ocorrem as etapas de seleção, cruzamento e mutação conforme os parâmetros definidos para o problema;
- 4ª Etapa (Avaliação): Os novos indivíduos gerados são avaliados através da função *fitness*;
- 5ª Etapa (Teste): Nesta etapa os critérios de parada são analisados onde caso não sejam atingidos retornará para a 3ª Etapa, caso contrário o processo é finalizado;
- 6ª Etapa (Fim): O processo é finalizado apresentando a melhor solução alcançada.



O pseudocódigo é apresentado na Figura 9. A execução tem o seu início na criação da população inicial representada pela letra “P”. Na segunda etapa, a população inicial é avaliada.

**Figura 9: Pseudocódigo do Algoritmo Genético**

```

1. inicializar população P; // inicia população de indivíduos
2. avaliação(P) // avaliar aptidão dos indivíduos
3. repita // critério de parada
4. selecione uma subpopulação P'; // candidatos a pais da próx. geração
5. para i ← 1 até taxa de cruzamento faça // cruzamento
6. escolha S1, S2 ∈ P'; //seleção
7. filho ← cruzamento (S1, S2);
8. se f(S1) ≥ f(S2) então Saux ← S1; // Maximização
9. senão Saux ← S2;
10. se f(Saux) ≥ f(filho) então
11. filho substitui Saux em P;
12. fim_se;
13. fim_para;
14. para i ← 1 até taxa de mutação faça; // mutação
15. selecione um cromossomo Sj em P;
16. Sj ← mutação(Sj);
17. fim_para;
18 até que critério_parada seja satisfeito;

```

**Fonte:** BRUM (2014).

Um laço de repetição se inicia até que o critério de parada seja alcançado. Na terceira etapa, é selecionado um grupo de indivíduos para gerar a próxima geração onde após é realizado o cruzamento e a mutação. Na etapa seguinte, a nova geração é avaliada e a iteração se repetirá até que os critérios de parada sejam satisfeitos (BRUM, 2014).

Os AGs podem ser utilizados para solucionar inúmeros problemas de otimização combinatória, dentre eles podemos exemplificar a Família de Partes (JEON; LEEP, 2006) e o Problema de *Scheduling* (MURATA; ISHIBUCHI, 1994). Porém, para resolver diferentes problemas, a modelagem e o conhecimento sobre a implementação do algoritmo é fundamental para se alcançar o objetivo. Durante a implementação, alguns conhecimentos se fazem necessários como: Geração da população inicial; Cromossomo; Função de avaliação; Processo de seleção; Operadores de Cruzamento e Mutação; Critérios de parada.

#### 3.4.1.1 População inicial

A inicialização do AG é a etapa em que os primeiros indivíduos são gerados. Esses indivíduos são soluções viáveis para o problema a ser otimizado, logo a criação da primeira população pode auxiliar o algoritmo a convergir em soluções eficazes, senão o mesmo pode não alcançar o objetivo ou aumentar o tempo necessário para gerar uma boa solução (GUIMARÃES, 2013).

Esta população é de um conjunto de cromossomos onde o tamanho da população pode variar de acordo com a instância do problema. A geração da população pode ser realizada de inúmeras formas: aleatoriamente, empregando uma heurística, através de dados reais partindo do resultado de outra metaheurística, entre outras alternativas (REEVES, 2003). Dentre os métodos empregados para a inicialização, um dos mais empregados é a geração aleatória pelo fato de conseguir fornecer de forma eficiente uma população inicial que se encontre na região viável para o problema (RENNER; EKART, 2003).

#### 3.4.1.2 Cromossomo

Cada cromossomo representa uma solução viável para o problema o qual está sendo solucionado utilizando um AG. A etapa de codificação irá definir como será a organização do cromossomo a fim de representar as informações da solução. Conforme o tipo do problema, os genes do cromossomo irão apresentar diferentes variáveis definidos pela Função Objetivo (FO) (HARFF, 2013).

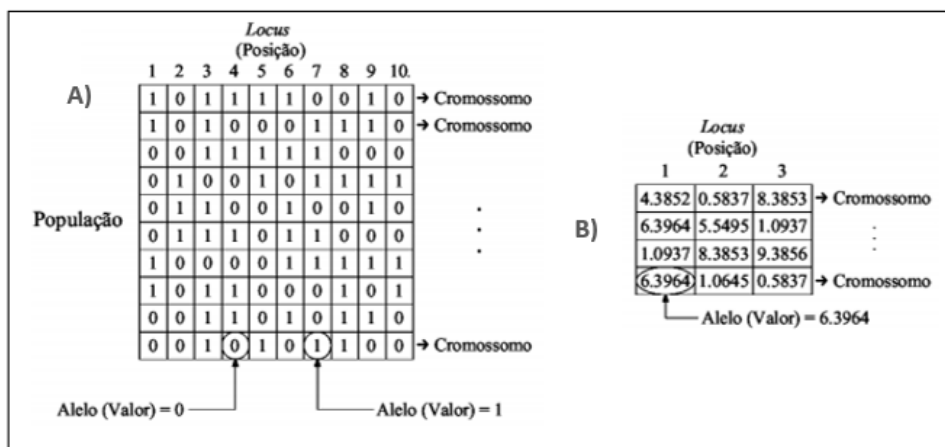
Uma das formas empregadas para a representação de um cromossomo é um vetor. Cada elemento do vetor irá representar uma variável de decisão para o problema. Os valores associados a cada gene poderão ser dos tipos: binários, inteiros, reais ou ainda uma combinação destes. (HARFF, 2013).

A codificação do cromossomo se torna mais simples com o uso de números binários, entretanto, em muitos problemas do mundo real, a representação binária pode não ser uma boa alternativa (DEB, 2011). Uma alternativa é a representação real, na qual emprega na representação pontos flutuantes, ou seja, números reais, normalmente utilizada em genes que estão definidos em intervalos contínuos (EIBEN; SMITH, 2003).

A Figura 10 apresenta duas formas empregadas para a representação dos cromossomos, sendo a matriz “A” a representação de cromossomos com valores binários, ou seja, somente aceita como possíveis valores em seus genes 0 ou 1 e a matriz “B” apresenta a representação através de números reais. Ao analisar as representações, independente dos problemas relacionados, é possível perceber que os cromossomos que são representados por números reais se tornam menores quanto aos valores que cada gene representa.

**Fonte:** Elaborada pelo autor.

**Figura 10: Representação de cromossomos binários e reais.**



### 3.4.1.3 Função Objetivo

A dinâmica do AG consiste em cada iteração gerar novos indivíduos partindo da população anterior, com a premissa de evoluir a cada iteração. Para que o algoritmo consiga avaliar os indivíduos gerados, é utilizada a Função Objetivo. Outros termos são utilizados como: Função de Avaliação, Função de Aptidão ou ainda o termo em inglês *Fitness*.

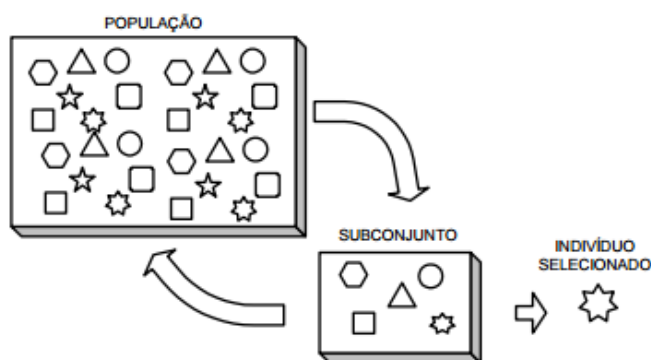
O principal objetivo da FO é analisar a aptidão de cada indivíduo da população dado o problema em análise. No caso de se tratar de um problema de maximização, os indivíduos de maior valor resultante da FO terão maior probabilidade de se manterem e serem escolhidos para gerar as futuras gerações. Em um problema de minimização, de maneira análoga ao problema anterior, os indivíduos de menor valor terão a maior probabilidade (POLTOSI, 2007).

### 3.4.1.4 Processo de Seleção

O Processo de Seleção visa determinar quais indivíduos de uma determinada geração serão mantidos ou quais serão selecionados como pais dos futuros indivíduos. Para a seleção, existem inúmeras técnicas para a escolha dos indivíduos, entre elas: Seleção por Torneio, Seleção Elitista, Seleção por Roleta, entre outros (REEVES, 2003; AL JADAAN; RAJAMANI; RAO, 2008).

A Seleção por Torneio uma parcela de indivíduos da população é selecionada aleatoriamente. Deste grupo, é realizado um processo de competição que utiliza o valor de aptidão e mecanismos randômicos para definir se irá selecionar o indivíduo com maior ou menor aptidão. Na Figura 11, é realizada uma abstração do método quanto a população, formação de uma subpopulação e a escolha de um indivíduo.

**Figura 11: Representação da seleção pelo método de torneio**

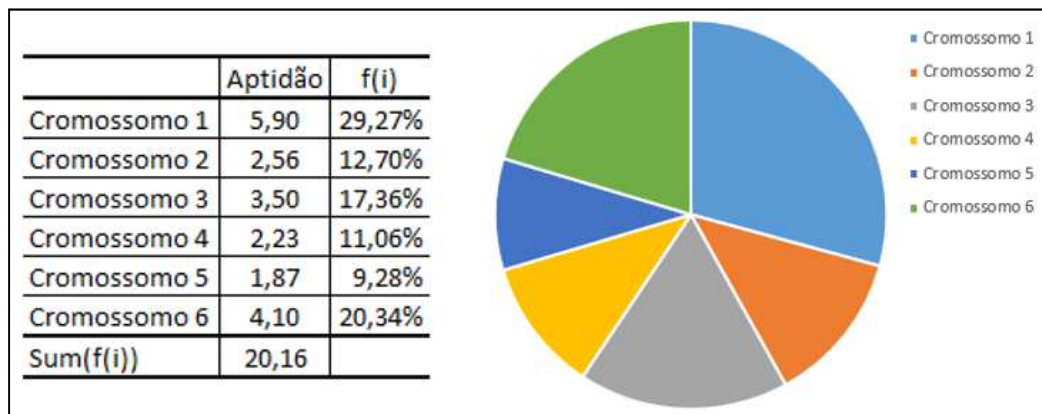


**Fonte:** Elaborada pelo autor.

A Seleção por Elitismo é uma técnica que visa melhorar o desempenho do AG pelo fato de parte de uma futura geração será criada simplesmente mantendo os indivíduos selecionados da população corrente, fazendo com que eles não sejam eliminados pelos processos de cruzamento ou mutação (NASCIMENTO, 2010).

De acordo com Pacheco (1999), na Seleção por Roleta cada indivíduo recebe uma porção da roleta proporcional a sua função de aptidão que é o resultado do seu valor de FO dividido pelo somatório do valor de FO de toda a população. Na Figura 12, é realizada uma simulação com uma população de seis cromossomos onde o somatório do valor de todas as aptidões resultou no valor de 20,16. Após o somatório, foi dividido o valor de aptidão de cada cromossomo gerando assim a sua fatia na roleta.

**Figura 12: Simulação da seleção por roleta.**



**Fonte:** Elaborada pelo autor.

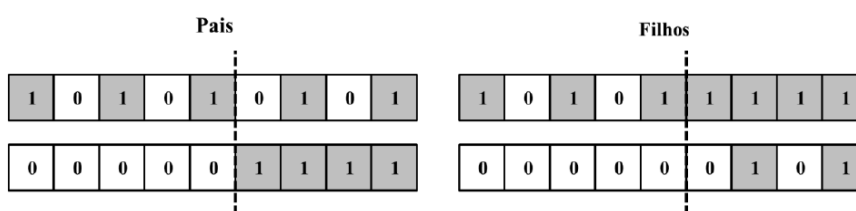
#### 3.4.1.5 Operadores de Cruzamento, Mutação e Epidemia

Conforme Harff (2013), o processo de cruzamento consiste em, de forma aleatória ou não, selecionar dois indivíduos para que a partir deles sejam gerados novos indivíduos onde estes serão novas soluções para o problema proposto, já que contêm parte de ambos os pais com a expectativa de apresentarem soluções melhores que as já alcançadas.

A operação que será realizada para a combinação dos genitores será particular de cada codificação, onde é possível encontrar inúmeras soluções na literatura, dentre elas é possível destacar: Cruzamento com ponto de corte único, Cruzamento com ponto de corte duplo, Cruzamento com pontos aleatórios entre outros (YALCINOZ; ALTUN, 2002).

No cruzamento com ponto de corte único, é selecionada uma localização predefinida ou aleatória, pelo algoritmo, no cromossomo dos pais, para assim realizar a divisão de cada um em duas partes, onde cada filho será composto pela união das partes de modo que cada novo indivíduo gerado receba informações genéticas dos dois pais, conforme é apresentado na Figura 13 (HOLLAND, 1975).

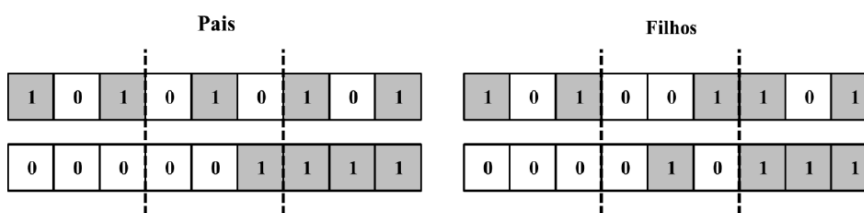
**Figura 13: Cruzamento com ponto de corte único.**



**Fonte:** Elaborada pelo autor.

O cruzamento com ponto de corte duplo gera indivíduos com uma maior diversidade genética, tendo em vista que ao invés de realizar a divisão de cada pai em dois, será selecionado um ponto adicional de corte, dividindo assim cada genitor em três partes conforme é ilustrado na Figura 14 (HOLLAND, 1975).

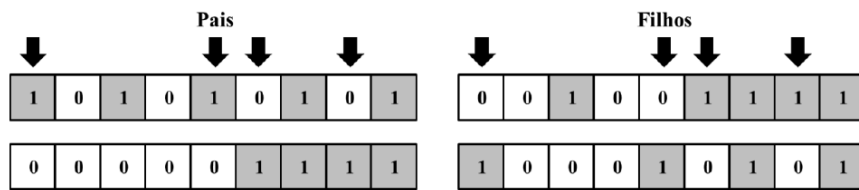
**Figura 14: Cruzamento com ponto de corte duplo.**



**Fonte:** Elaborada pelo autor.

Levando em consideração a existência de inúmeras possíveis formas de cruzamento, é possível destacar, por fim, o cruzamento de pontos aleatórios, no qual a maneira com que os genitores serão partidos ocorrerá de forma dinâmica podendo variar a quantidade e a posição onde cada ponto de corte será posicionado, conforme é apresentado na Figura 15 (HOLLAND, 1975).

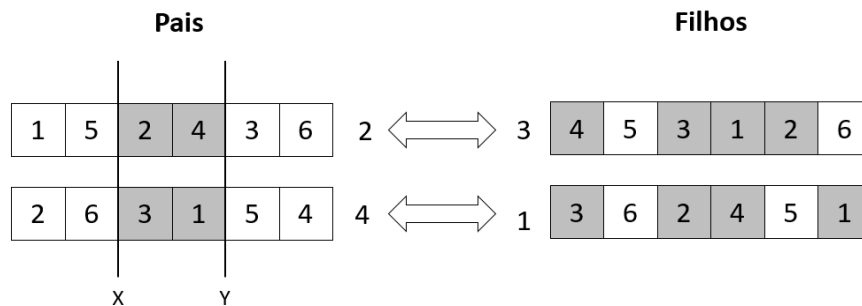
**Figura 15: Cruzamento com pontos aleatórios**



Fonte: Elaborada pelo autor.

O PMX é uma das primeiras soluções apresentadas com a finalidade de resolver o problema de inviabilidade das soluções. Nesta técnica, dois pontos são selecionados para definir o mapeamento de troca que será realizado entre os cromossomos (GALAFASSI, 2011). A Figura 16 apresenta um exemplo do cruzamento utilizando PMX, onde dois pontos denominados X e Y definem a forma que foram trocados os genes dos cromossomos.

**Figura 16: Cruzamento utilizando PMX**



Fonte: Elaborada pelo autor.

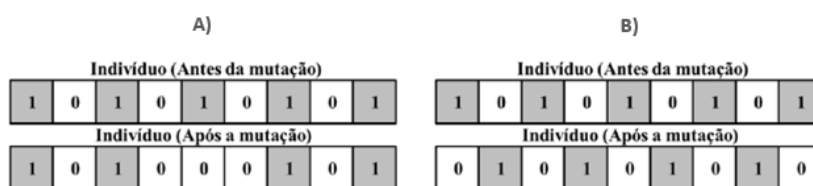
É importante salientar a importância do cruzamento, principalmente quanto às taxas a serem utilizadas, amplificando assim as buscas por melhores soluções. Na literatura se encontram valores que podem variar entre 70% e 90%. Em seu trabalho, Harff (2013) utiliza um percentual em torno de 90% para alcançar os seus objetivos. Já Guimarães (2013) utilizou 80% para conseguir alcançar uma boa solução. Entretanto, a amplitude entre os intervalos para

a taxa de cruzamento pode ser maior segundo Andrade et al. (2008), podendo variar de 50% até 100%.

O AG possui um mecanismo para a diversificação das buscas, este mecanismo recebe a terminologia de Mutação. A Mutação é responsável por garantir que novos cromossomos sejam gerados com alguma característica que não tenha sido herdada de seus genitores, evitando assim a falta de diversidade dos indivíduos ao longo das gerações. De acordo com Nascimento (2010), esta mutação deve ser de forma aleatória, entretanto o novo valor deve estar contido dentro de uma faixa pré-estabelecida com probabilidade de ocorrência para que a otimização não se torne um fator puramente aleatório.

A Figura 17 apresenta a realização da Mutação em dois cromossomos de formas diferentes. Sendo uma delas apenas com a inversão de um ponto e a outra realizando a alteração de todos os valores.

**Figura 17: Mutação em (a) apenas um gene e no cromossomo (b) todos os genes**



**Fonte:** Elaborada pelo autor.

Com a finalidade de diversificar as buscas, o percentual de Mutação a ser aplicado é fator importante e vai variar de acordo com o problema a ser solucionado. Na literatura, em diversos trabalhos são encontrados percentuais baixos de mutação em torno de 1% conforme aplicado por Guimarães (2013) em seu trabalho, corroborando com Czarkowski (2002) que sugere percentuais entre 0,1% até 1%. Entretanto, não pode virar via de regra já que esta taxa vai variar conforme o problema a ser abordado, como por exemplo, no trabalho de Brum (2014) foram empregados percentuais entre 10% e 15%, outro caso extremo foi o problema que Harff (2013) tratou, onde obteve uma melhor solução em um menor tempo utilizando 90% como parâmetro de mutação.

A Epidemia é realizada com base em dois parâmetros definidos previamente: (i) define quando ela deve ocorrer, baseada no número de gerações consecutivas sem melhora e (ii) o percentual da população a ser substituída quando o processo for ativado. Uma vez ativado, o operador de Epidemia realiza a substituição dos indivíduos utilizando a mesma estratégia



para a geração da população inicial. Ao final do processo Epidêmico a nova população é devolvida para fluxo normal do AG (CHIWIACOWSKY; MONTEIRO, 2012).

#### 3.4.1.6 Critério de parada

O critério de parada tem por finalidade definir quando o algoritmo deve parar para evitar um processamento por tempo indefinido. Existem inúmeros critérios que podem ser aplicados para determinar o fim da execução, entre eles podemos destacar: o número máximo de gerações, um valor mínimo da FO, tempo computacional, diversidade da população entre outros (REEVES, 2003). Harff (2013) e Link (2012) empregaram em seus trabalhos o número máximo de gerações enquanto Brum (2014) utilizou o critério de número máximo de gerações desde que não obtivesse melhora expressiva no resultado da FO.

### 3.5 Trabalhos relacionados

Para a busca de trabalhos com o mesmo tema, foi realizada uma busca no site *ScienceDirect* que é uma plataforma operada pela editora anglo-holandesa Elsevier, que está em funcionamento desde 1997. Esta plataforma provê acesso a aproximadamente 2500 revistas científicas e a mais de 26000 *books*. A pesquisa foi realizada no site utilizando 23 combinações. As combinações utilizadas levaram em consideração o contexto no qual este trabalho se enquadra. Abaixo, segue a Tabela 3 apresentando os resultados da pesquisa.

**Tabela 3: Resultado das pesquisas**

palavras chaves	resultados	ano inicial	ano final	diferença	média/ano
warehouse	45323	1824	2017	193	234,83
"warehouse layout"	178	1975	2017	42	4,24
"warehouse layout problem"	16	1982	2017	35	0,46
"order picking"	591	1949	2017	68	8,69
warehouse AND "order picking"	484	1949	2017	68	7,12
"warehouse layout" AND "order picking"	100	1975	2017	42	2,38
"warehouse layout problem" AND "order picking"	9	1982	2017	35	0,26
"genetic algorithms"	37481	1974	2017	43	871,65
warehouse AND "genetic algorithms"	1196	1985	2017	32	37,38
"warehouse layout" AND "genetic algorithms"	32	1997	2017	20	1,60
"warehouse layout problem" AND "genetic algorithms"	7	2006	2014	8	0,88
"order picking" AND "genetic algorithms"	63	1992	2017	25	2,52
warehouse AND "order picking" AND "genetic algorithms"	54	1992	2017	25	2,16
"warehouse layout" AND "order picking" AND "genetic algorithms"	15	2005	2017	12	1,25
"warehouse layout problem" AND "order picking" AND "genetic algorithms"	2	2008	2011	3	0,67
metaheuristic	15441	1976	2017	41	376,61
"order picking" AND metaheuristic	93	1996	2017	21	4,43
warehouse AND metaheuristic	1111	1994	2017	23	48,30
"warehouse layout" AND metaheuristic	38	1997	2017	20	1,90
"warehouse layout problem" AND metaheuristic	3	2006	2014	8	0,38
warehouse AND "order picking" AND metaheuristic	81	2005	2017	12	6,75
"warehouse layout" AND "order picking" AND metaheuristic	26	2005	2017	12	2,17
"warehouse layout problem" AND "order picking" AND metaheuristic	1	2008	2008	1	1,00

**Fonte:** Elaborada pelo autor.

Após o levantamento quantitativo da pesquisa realizada pelos termos relacionados ao contexto do trabalho, foi realizado um levantamento dos trabalhos que fazem referência de alguma forma à palavra-chave “*warehouse layout problem*”. A Tabela 4 apresenta os trabalhos que referenciam o problema.

**Tabela 4: Trabalhos retornados para “warehouse layout problem”**

Autores	Ano	Título
Guoqing Zhang, Tatsushi Nishi, Sarina D.O. Turner, Keisuke	2017	An integrated strategy for a production planning and warehouse layout problem: Modeling and solution approaches
Miguel Horta, Fábio Coelho, Susana Relvas	2016	Layout design modelling for a real world just-in-time warehouse
R. Kia, F. Khaksar-Haghani, N. Javadian, R. Tavakkoli-Mogha	2014	Solving a multi-floor layout design model of a dynamic cellular manufacturing system by an efficient genetic algorithm
A. Azadeh, S. Motevali Haghighi, S.M. Asadzadeh	2014	A novel algorithm for layout optimization of injection process with
Dragan Matić, Jozef Kratica, Vladimir Filipović, Djordje Dugc	2012	Variable neighborhood search for Multiple Level Warehouse Layout Problem, Electronic Notes in Discrete Mathematics
Yanru Chen, Qinxin Xiao, Xiaoping Tang	2011	Product Layout Optimization and Simulation Model in a Multi-level Distribution Center
Avninder Gill	2009	Determining loading dock requirements in production-distribution facilities under uncertainty
Peter Baker, Marco Canessa	2009	Warehouse design: A structured approach
Semih Önüt, Umur R. Tuzkaya, Bilgehan Doğaç	2008	A particle swarm optimization algorithm for the multiple-level warehouse layout design problem
Suk-Chul Rim, In-Sun Park	2008	Order picking plan to maximize the order fill rate, Computers &
G.Q. Zhang, K.K. Lai	2006	Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem
J. Ashayeri, R. Heuts, B. Tammel	2005	A modified simple heuristic for the p-median problem, with facilities design applications
Thomas L. Landers, Melinda K. Beavers, Malik Sadiq, Don E	1994	Software for dynamic reconfigurable order picking systems
Mickey R. Wilhelm, Robert A. Lichtefeld, Thomas L. Ward	1986	Computer-aided facility location
Charles J. Malmborg, Swaminathan Balachandran, Daniel M	1986	A model based evaluation of a commonly used rule of thumb for warehouse layout, Applied Mathematical Modelling
Jessica O. Matson, John A. White	1982	Operational research and material handling

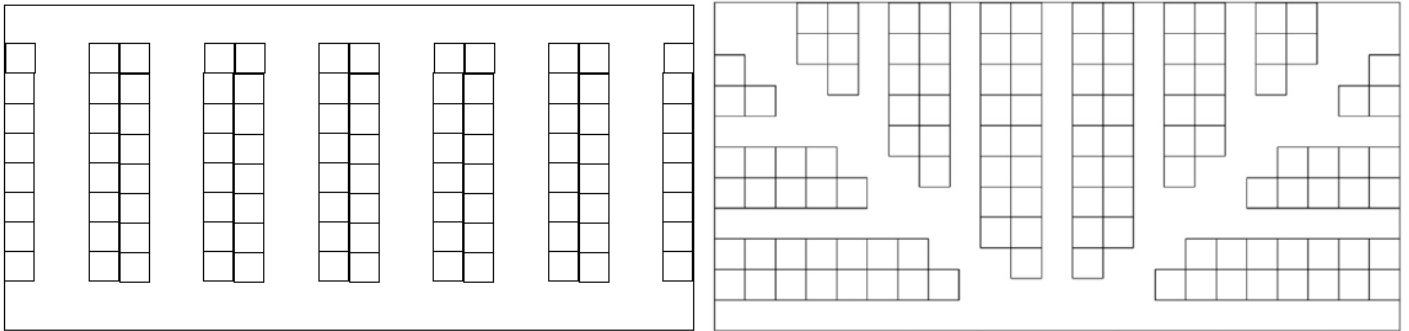
**Fonte:** Elaborada pelo autor.

O trabalho “*Layout design modelling for a real world just-in-time warehouse*” aplicou um estudo com a finalidade de gerar um *layout* que otimizasse a operação de *crossdocking* (operação na qual os produtos são recebidos, separados e encaminhados a outro veículo sem a necessidade de armazenar). O estudo foi aplicado em um CD de alimentos portugueses onde comprovou que é possível reduzir em 2000 km por mês o deslocamento das mercadorias de um veículo para o outro (HORTA et al., 2016).

Apesar de sugerir um *layout* diferente do tradicional, o trabalho “*Detailed design of fishbone warehouse layouts with vertical travel*” criou um método para gerar um *layout* tridimensional onde os corredores são na diagonal, similar a uma espinha de peixe. O objetivo do trabalho era reduzir a dependência de especialistas no projeto do armazém. Neste trabalho, foi criado um algoritmo para a geração do *layout* inicial, onde após a geração foi empregado o AG para a otimização. Como resultado em 91,74% dos casos ficaram próximos da melhor solução (CARDONA et al., 2015).

Para melhor compreensão da diferença dos *layouts*, abaixo segue a Figura 18, onde à esquerda está o *layout* tradicional, caracterizado por todos os corredores estarem paralelamente em relação a lateral do armazém, enquanto o *layout* da direita espinha de peixe, possui corredores na diagonal (CARON; MARCHET; PEREGO, 2000; CARDONA et al., 2015).

**Figura 18:** *Layout* tradicional e espinha de peixe



**Fonte:** Elaborada pelo autor.

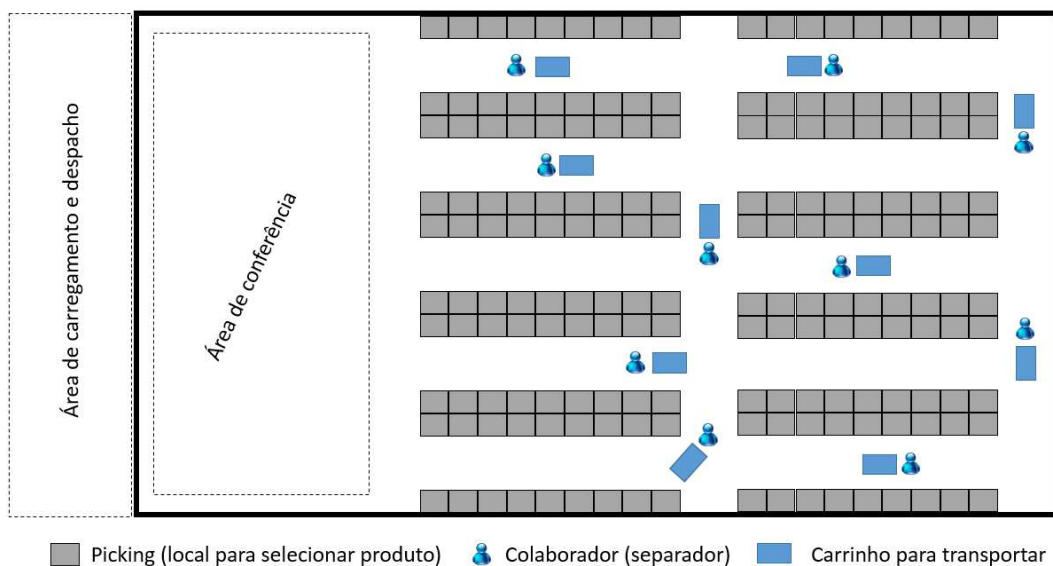
## 4 MODELO

Este capítulo apresenta o modelo deste trabalho. A seção 4.1 descreve o ambiente em que o modelo se insere, na seção 4.2 estão as regras que definem o problema, conforme as informações obtidas através da revisão literária. A formulação desenvolvida encontra-se na seção 4.3. A arquitetura da aplicação é detalhada na seção 4.4 enquanto as tecnologias aplicadas para o desenvolvimento são apresentadas na seção 4.5.

### 4.1 Apresentação do Ambiente

Apesar da automatização, Napolitano (2012) afirma que cerca de 80% da seleção de pedidos são realizados por colaboradores que caminham pelo depósito utilizando um carrinho a fim de recuperar as mercadorias das prateleiras. O armazém descrito neste trabalho emprega um *layout* tradicional, caracterizado pela disposição dos corredores paralelos e ortogonal em relação às paredes (CARON; MARCHET; PEREGO, 2000). A Figura 19 elucida o formato de seleção manual de pedidos considerada nesta modelagem.

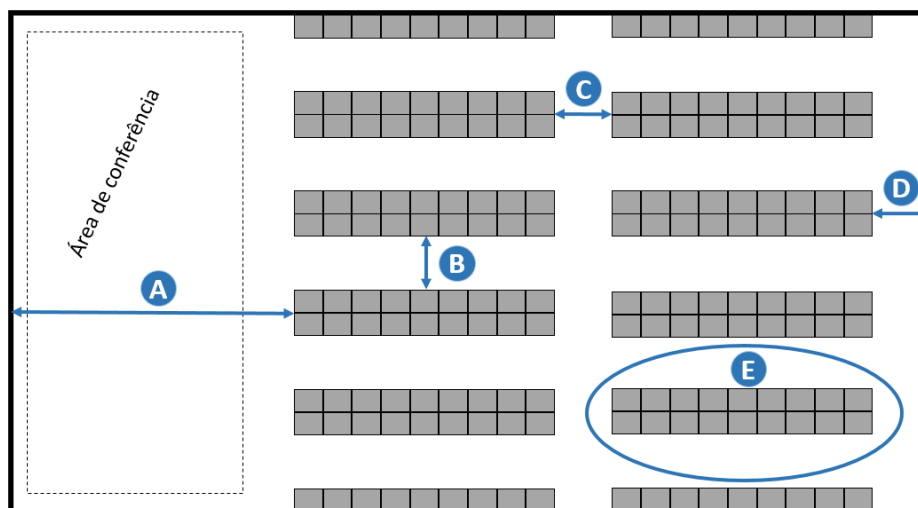
**Figura 19: Seleção manual de pedidos**



**Fonte:** Elaborada pelo autor.

O *layout* do armazém deve considerar os espaços necessários para a movimentação das mercadorias e a circulação dos colaboradores. Estes espaços serão considerados a fim de gerar um *layout* viável de aplicação em um CD. Os espaços para a movimentação das mercadorias são apresentados na Figura 20.

**Figura 20: Espaços para a movimentação das mercadorias**



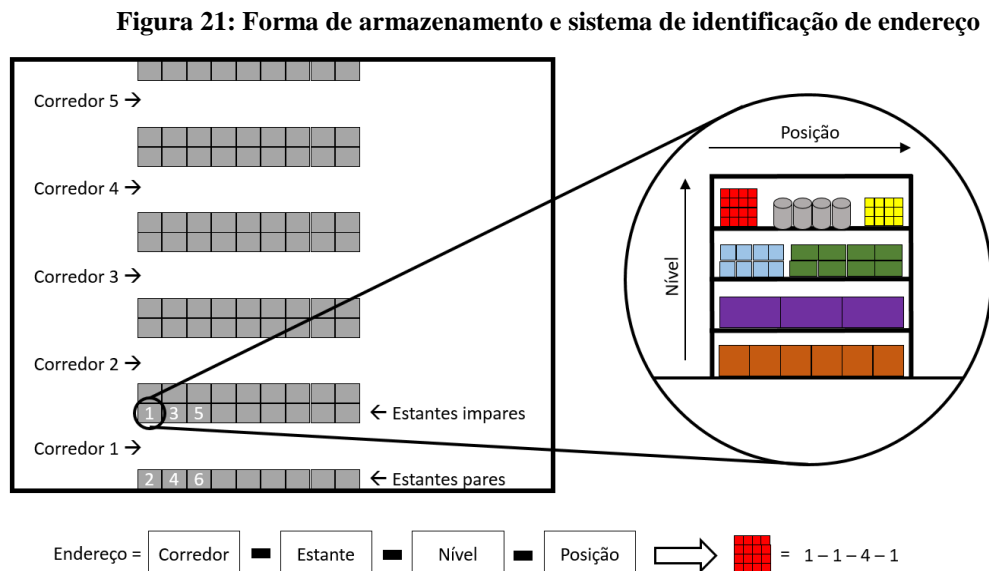
**Fonte:** Elaborada pelo autor.

Abaixo, segue a definição de cada espaço apresentado na Figura 20:

- a) A – Espaço para a conferência das mercadorias separadas;
- b) B – Espaço entre as estantes para permitir a circulação dos colaboradores;
- c) C – Espaço entre os blocos de estantes para melhorar a circulação;
- d) D – Espaço entre o final do bloco e a parede para melhorar a circulação;
- e) E – O número de estantes por bloco.

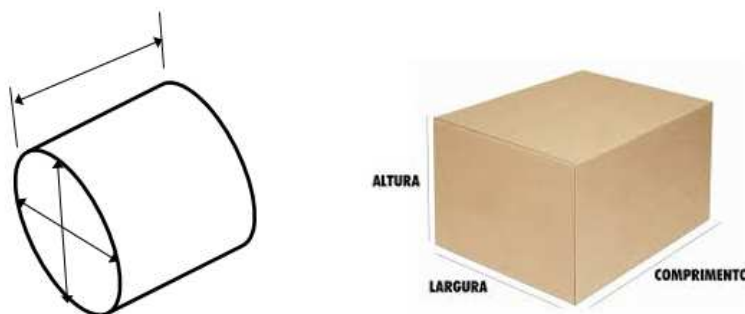
O armazenamento das mercadorias ocorre em estantes (estruturas metálicas) agrupadas em blocos e separadas por corredores. Para cada mercadoria em um CD, existirá no mínimo um endereço de armazenagem, para este endereço existirá um padrão de endereçamento. Este deve ser simples, lógico e o mais específico possível, identificando cada um dos elementos dos locais de armazenagem. A primeira posição indica o corredor, a segunda refere-se à estante, a terceira indica o nível (a partir do nível do chão no sentido ascendente). A última, indica a posição na

prateleira no sentido da esquerda para a direita (em cada nível haverá no mínimo um endereço) conforme é apresentado na Figura 21 (ACKERMAN, 1997). Um cuidado importante é quanto ao peso das mercadorias associadas em cada nível, onde deve-se evitar mercadorias pesadas em alturas evitando assim possíveis acidentes ou lesões aos colaboradores.



**Fonte:** Elaborada pelo autor.

As mercadorias possuem inúmeras formas. Um parâmetro utilizado é a cubagem para calcular o espaço de armazenamento necessário. A cubagem é o resultado da multiplicação das medidas (altura, largura, comprimento). A Figura 22 apresenta o cálculo realizado através da multiplicação das medidas (altura x largura x comprimento) denominado cubagem, que é o espaço considerado para o armazenamento de uma mercadoria, onde apesar do formato cilíndrico o espaço considerado será o mesmo de uma caixa.

**Figura 22: Cubagem das mercadorias**

**Fonte:** Elaborada pelo autor.

Foram apresentados nesta seção as características quanto à forma de coleta e a forma de armazenamento das mercadorias. A seção 5.2 irá relacionar as características do ambiente apresentadas com as regras que serão consideradas para o modelo.

#### **4.2 Definição das regras do Modelo**

O objetivo deste trabalho é gerar uma aplicação capaz de criar um *layout* aplicável em um armazém com a finalidade de reduzir os custos no processo de seleção de pedidos. Nesta seção serão apresentadas as regras que irão ajudar na formulação matemática do problema.

A seção 4.1 apresentou os detalhes do ambiente, destacando as características necessárias para a extração das regras que irão compor o modelo. A Figura 19 apresentou o modelo de seleção de pedidos que é manual e a Figura 20 complementou com as particularidades do ambiente onde destacam-se:

- a) Espaço para a conferência: É uma área destinada para a realização da conferência dos produtos, ou seja, este espaço será considerado como reservado ou ocupado;
- b) Corredor: É o espaço necessário para a circulação dos colaboradores com os seus carrinhos para a retirada das mercadorias das prateleiras para a entrega na área de conferência;
- c) Espaço entre os blocos: A fim de evitar corredores extensos estes espaços tem a finalidade de melhorar a circulação;



- d) Espaço entre o final do bloco e a parede: Similar ao corredor, melhorará a circulação dos colaboradores, viabilizando o movimento em formato de “U” onde pode ir por um corredor e retornar por outro corredor sem a necessidade de retornar todo o caminho já percorrido;
- e) Número de estantes por bloco: Este parâmetro tem a finalidade de dimensionar os blocos que serão gerados pela aplicação;
- f) Tamanho do armazém (comprimento e largura): Apesar de implícito na Figura 19, são medidas importantes para a definição do *layout*, já que define os limites;
- g) Largura da estante: Apesar de implícita na Figura 19 é um parâmetro fundamental para a definição do número de corredores e blocos.

Após regras referentes ao ambiente, outras entidades do ambiente começam a ser apresentadas (Figura 21), a estante, o nível e a posição onde:

- a) A estante: É a estrutura metálica que irá comportar os níveis. Uma estante terá no mínimo um nível;
- b) O nível: Trata-se de uma espécie de prateleira que irá conter no mínimo uma posição. O nível possui uma restrição quanto ao peso, ou seja, a soma de todas as mercadorias que estarão sobre ele não poderá ultrapassar o limite estabelecido;
- c) A posição: É o local onde a mercadoria será depositada, uma posição está relacionada somente a uma mercadoria.

A última entidade a ser apresentada é a mercadoria (Figura 22) quanto às características de dimensionamento. A mercadoria neste modelo passa a desempenhar o papel fundamental, pois o *layout* será montado com base nas características físicas e de negócio:

- a) Físicas: Essas características irão definir o espaço que será necessário para o armazenamento. Essas características são: altura, largura, comprimento;
- b) Negócio: São características geradas a partir das informações relacionadas à comercialização das mercadorias. Neste modelo esta característica é o número médio de solicitações de retirada das mercadorias do endereço.

O pedido é definido pela solicitação de um cliente de uma ou mais mercadorias, a este é atribuído um identificador único, normalmente numérico. As solicitações de coleta estão diretamente relacionadas ao número de pedidos que as mercadorias são relacionadas, ou seja,

se uma determinada mercadoria aparece em dez pedidos esta terá o mesmo número de solicitações de coleta.

Esta seção apresentou as regras relacionadas às características do ambiente apresentado na seção anterior. A seção 4.3 apresenta a definição matemática para o modelo.

### 4.3 Formulação Matemática

Nesta seção, serão apresentadas as formulações matemáticas para o ambiente e regras definidas nas seções 4.1 e 4.2. Outro objetivo desta seção é apresentar a função objetivo. Abaixo, seguem as entidades relacionadas ao *layout*.

- a)  $C$  = Comprimento do armazém;
- b)  $L$  = Largura do armazém;
- c)  $CC$  = Comprimento da área de conferência;
- d)  $LE$  = Largura das estantes;
- e)  $CE$  = Define o comprimento das estantes;
- f)  $LC$  = Largura mínima dos corredores e espaço entre os blocos;
- g)  $LF$  = Largura entre o final do bloco e parede;
- h)  $NE$  = Número de estantes por bloco;
- i)  $c$  = Define os corredores;
- j)  $b$  = Define os blocos;
- k)  $be$  = Define os espaços entre os blocos;
- l)  $e$  = Define as estantes;
- m)  $ed$  = Distância relativa entre a estante e a área de conferência;
- n)  $n$  = Define os níveis;
- o)  $p$  = Define uma posição;
- p)  $m$  = Define as mercadorias
- q)  $mc$  = Define a cubagem da mercadoria;

r)  $md$  = Média diária de solicitações para a retirada da prateleira.

Formulação da proposta é:

$$\text{Minimizar} = \sum_{i=1}^e \sum_{j=1}^n \sum_{k=1}^p ed_i * md_{ijk} \quad (5.1)$$

Sujeito a

$$C, L, CC, LE, LC, LF, n_k, be, ed, nk, nc, mk, mc \in \mathbb{R}^* \quad (5.2)$$

$$NE, NR, r, c, b, e, n, p \in \mathbb{N} \quad (5.3)$$

$$\sum_{i=1}^c c_i * LC + \sum_1^e e_1 * LE + \sum_2^e e_2 * LE \leq L \quad (5.4)$$

$$\left( \sum_1^c \sum_{j=1}^e e_j * CE \right) + \left( \sum_1^c \sum_{i=1}^{be} be_i * LC \right) + CC + LF \leq C \quad (5.5)$$

$$\sum_{i=1}^n \sum_{j=1}^p mc_{ij} \leq nc_i \quad (5.6)$$

O objetivo do trabalho é minimizar o custo do layout, onde para calcular o custo realiza-se o somatório da posição de todos os produtos alocados multiplicados pela média dos pedidos por dia conforme é apresentado na fórmula (5.1).

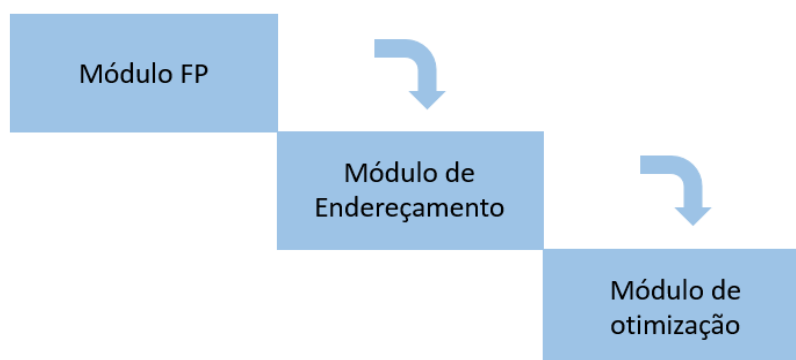
As restrições apresentadas nas fórmulas (5.4) e (5.5) fazem referência aos limites do armazém quanto à largura e ao comprimento. A restrição (5.6) expressa o limite a ser respeitado no momento da alocação do produto, que a cubagem total dos produtos alocados devem ser menor ou igual a cubagem disponível para alocação.

#### 4.4 Arquitetura da aplicação

Após as definições realizadas nas seções anteriores, esta seção possui a finalidade de apresentar a arquitetura do protótipo construído. As informações necessárias foram extraídas

da revisão literária que auxiliou a entender as funcionalidades necessárias para alcançar o objetivo. A Figura 23 apresenta a modularização que será desenvolvido na aplicação.

**Figura 23: Módulos do protótipo**



**Fonte:** Elaborada pelo autor.

O módulo FP terá a finalidade de analisar a lista de mercadorias e, conforme a similaridade, gerar subconjuntos a partir das características de cada mercadoria, estas características são apresentadas na Tabela 5.

**Tabela 5: Dados de entrada referente as mercadorias**

Mercadoria	
Propriedade	Descrição
Código	Identificador do produto
Descrição	Texto que descreve o produto
Altura	Altura do produto
Largura	Largura do produto
Comprimento	Comprimento do produto
Giro dia	Define a quantidade média vendida por dia
Pedido dia	Quantidade média de pedidos por dia

**Fonte:** Elaborada pelo autor.

A lista de mercadorias é inicializada, após a posição de cada mercadoria é alterada dinamicamente, garantindo assim a aleatoriedade na criação das FPs. A criação é realizada em duas fases, onde na primeira fase é realizado o agrupamento das mercadorias por similaridade de dimensões, enquanto na segunda fase é realizado uma união das FPs que utiliza o mesmo espaço de armazenamento. A Figura 24 apresenta o pseudocódigo para a criação das FPs.

**Figura 24: Pseudocódigo para a criação das FPs.**

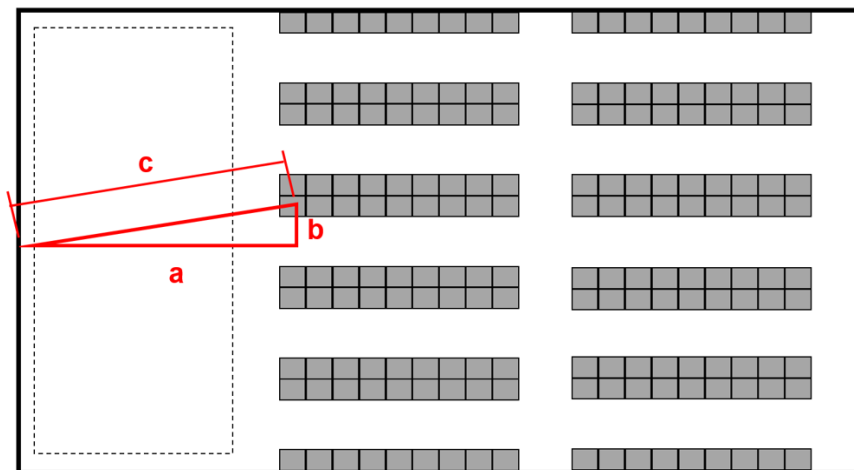
```

inicializar lista; // carrega a lista de mercadorias
embaralhar (lista); // altera a ordem das mercadorias na lista
para cada mercadoria faça
    classificaMercadoriaFP (mercadoria); // realiza a classificação
fim;
calcularArmazenamento(familiasParte); // calcula o espaço de armazenamento para cada FP
otimizarFamilias(familiasParte); // realiza a união baseado no espaço de alocação
  
```

**Fonte:** Elaborada pelo autor.

O Endereçamento é responsável por alocar cada mercadoria em uma determinada estante. O processo é iniciado após a geração de uma lista contendo todas as estantes ordenadas pela distância em relação ao ponto central na frente do armazém. Esta distância é calculada através da trigonometria achando-se o valor da hipotenusa do triângulo retângulo formado entre a posição da estante e o ponto de referência, conforme é apresentado na Figura 25 ilustrando a formação do triângulo retângulo.

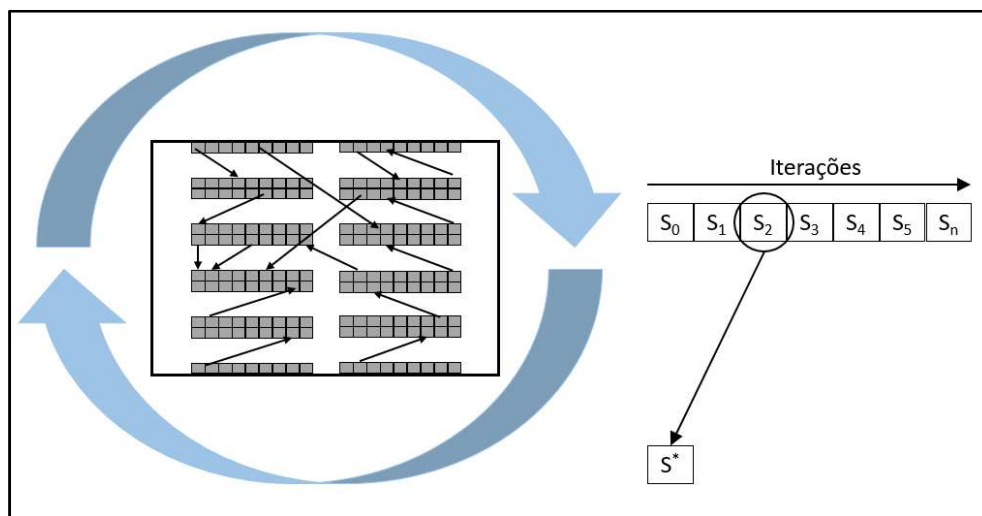
**Figura 25: Distribuição das mercadorias nas estantes**



**Fonte:** Elaborada pelo autor.

Ao final da execução do módulo de Endereçamento, têm-se inúmeras soluções para o problema. A próxima etapa é iniciar a melhora das soluções através do módulo de Otimização que terá a função de alterar a localização das mercadorias a fim de gerar soluções mais eficientes. A Figura 26 representa as iterações que serão realizadas pelo módulo de Otimização que, a cada iteração, irá selecionar a melhor solução simbolizado por  $S^*$ , onde irá manter a melhor solução encontrada até que os critérios de parada sejam alcançados.

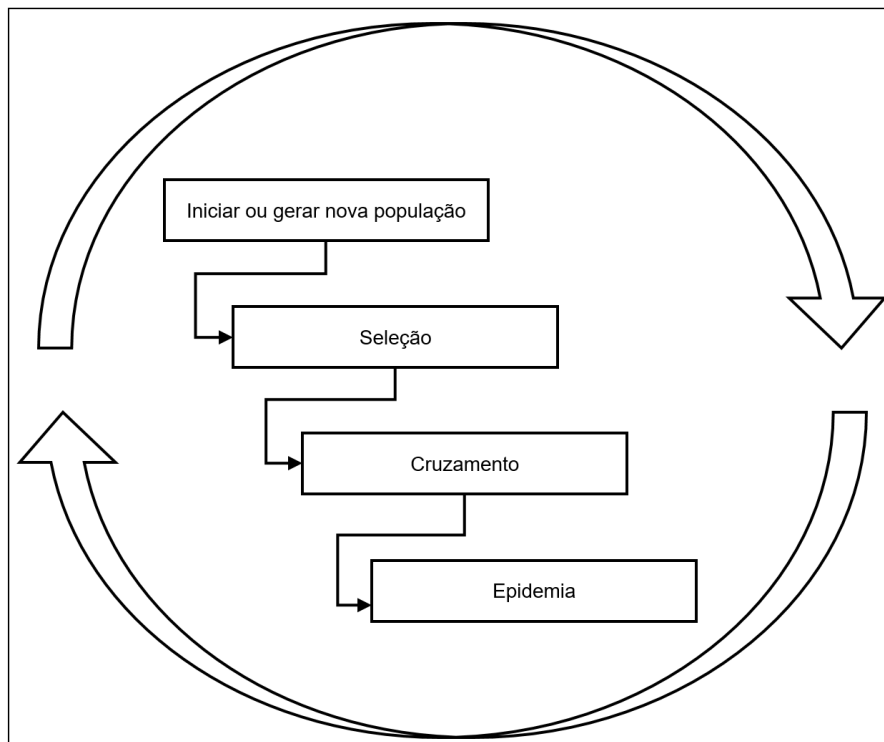
**Figura 26: Módulo de otimização**



**Fonte:** Elaborada pelo autor.

Para gerar as novas soluções, o módulo de Otimização utiliza o AG. A Figura 27 apresenta fluxograma dos processos aplicados, onde o processo é inicializado através da geração de uma população inicial, onde após é aplicado o processo de seleção, cruzamento e por último o processo de epidemia. Quando o critério de parada é alcançado, no caso deste trabalho se trata do número de gerações, é retornada a melhor solução encontrada.

**Figura 27: Fluxo do processo de otimização**



**Fonte:** Elaborada pelo autor.

#### 4.5 Tecnologias aplicadas

Esta seção descreverá os detalhes das tecnologias empregadas para o desenvolvimento do protótipo, iniciando pela engenharia de *software*, arquitetura, linguagem de programação, tecnologias de armazenamento como banco de dados e finalizando com as tecnologias empregadas no desenvolvimento da interface.

#### 4.5.1 Engenharia de *Software*

O *software* se torna cada vez mais necessário e parte do cotidiano, mesmo sem percebermos quando o utilizamos, como por exemplo, um micro-ondas. A elaboração de um *software* vem se tornando cada vez mais complexa, devido ao número de pessoas e pelos recursos envolvidos, onde antes eram desenvolvidos por uma única pessoa, hoje é desenvolvido por grandes equipes devido ao aumento da complexidade.

Conforme o investimento em uma aplicação aumenta, a expectativa de sua longevidade cresce na mesma proporção. Com isso, a necessidade de manutenção, adaptação ou melhorias se tornam vitais nos ciclos de vida de uma aplicação, onde devido a este fato se torna imprescindível que o desenvolvimento de uma aplicação deve passar pelos processos de engenharia (PRESSMAN, 2011).

Engenharia de *software* é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de *software*, desde os estágios iniciais da especificação do sistema até a sua manutenção, quando o sistema já está sendo usado. (SOMMERVILLE, 2011).

Não existe uma metodologia ou técnica universais que seja adequada a todos os sistemas e empresas de desenvolvimento de *software*, mas sim ferramentas e métodos que vêm evoluindo nas últimas décadas.

De forma geral, engenheiros de *software* aplicam métodos, ferramentas e teoria em busca de soluções para problemas, entretanto na engenharia o foco não é somente com os processos de desenvolvimento de *software*, ela também inclui atividades como gerenciamento de projeto, desenvolvimento de ferramentas, teorias e métodos para auxiliar na produção de *software*. (SOMMERVILLE, 2011).

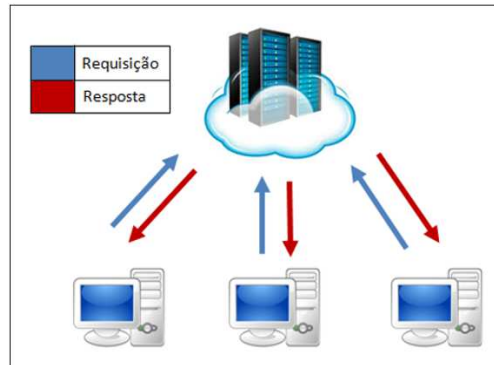
#### 4.5.2 Arquitetura da aplicação quanto à tecnologia

A aplicação terá uma estrutura Cliente-Servidor que é um modelo que possibilita que inúmeros computadores troquem informações. Neste modelo, existe um computador ou inúmeros computadores centralizando as requisições dos demais computadores. O fluxo de



troca de informações normalmente parte com uma requisição do cliente que é recebida pelo servidor, que executa o processamento das solicitações gerando uma resposta contendo o resultado para o cliente, conforme é possível ver na Figura 28.

**Figura 28: Representação da arquitetura Cliente-Servidor**



**Fonte:** Elaborada pelo autor.

A comunicação entre os clientes e os servidores ocorre através da pilha de protocolos TCP/IP (*Transmission Control Protocol-Internet Protocol*) utilizando o protocolo HTTP (*Hypertext Transfer Protocol*). A aplicação cliente normalmente é um navegador (Internet Explorer, Firefox, Chrome entre outros) que recebe o retorno do servidor onde são responsáveis por interpretar a resposta para o usuário (GORNI et al., 2007).

A aplicação desenvolvida utiliza como principal recurso de comunicação o protocolo *HTTP*, onde o servidor se encarrega de acessar as informações solicitadas no banco de dados e devolve páginas que são interpretadas pelos navegadores conforme a Figura 29.

**Fonte:** Elaborada pelo autor.

**Figura 29: Arquitetura da aplicação**



Nas próximas seções, serão apresentadas as tecnologias aplicadas para o desenvolvimento da aplicação desde a busca da informação no banco de dados até a apresentação para o usuário.

#### 4.5.3 Linguagem de programação

A linguagem de programação escolhida para o desenvolvimento da aplicação foi o Java. O Java teve seu início em 1991, na Sun Microsystems onde tinha o objetivo de integrar aparelhos domésticos e computadores. Entretanto, o uso do Java como solução *Web* teve seu marco em 1995 em um encontro do setor onde teve suas atenções voltadas para o Java devido ao foco em programação *Web* (DEITEL; DEITEL, 2010).

Melo (2010) enfatiza que Java teve como base em seu desenvolvimento um subconjunto da linguagem C++. Com a larga utilização no desenvolvimento de aplicações para a *Internet*, tornou-se rapidamente popular.

Segundo Horstmann e Cornell (2010), o Java desde a sua criação nunca foi uma linguagem qualquer. Atualmente, existe uma infinidade de linguagens de programação disponíveis, porém poucas têm o mesmo poder do Java. É uma plataforma de desenvolvimento com uma imensa biblioteca propiciando a reutilização de código reduzindo assim o esforço na criação de uma aplicação, fornece um ambiente de execução seguro e uma portabilidade para inúmeros sistemas operacionais.

A estrutura de programas e classes em Java segue a organização de linguagens tradicionais como C e C++, mas sem elementos que tornam programas e programação mais complexos. Após o aprendizado dos conceitos básicos de programação orientada a objetos, o estudante da linguagem pode começar a criar aplicativos úteis e complexos. A simplicidade se reflete também na maneira com que arquivos contendo programas em Java são compilados e executados: se as recomendações básicas forem seguidas, o compilador se encarregará de compilar todas as classes necessárias em uma aplicação automaticamente, sem necessidade de arquivos adicionais de configuração e inclusão de bibliotecas. (SANTOS, 2003).

A Programação Orientada a Objetos (POO) é um paradigma de programação que tem dominado atualmente o mundo da programação, substituindo a programação estruturada criada nos meados dos anos de 1970 (HORSTMANN; CORNELL, 2010).

Moreira Neto (2009) destaca que o modelo orientado a objetos usa como base a execução de métodos (são os dados de um objeto manipulados por pequenas funções),

viabilizando desenvolver aplicações mais próximas de como o usuário visualiza os sistemas e suas funcionalidades.

Por causa das diferenças estruturais e conceituais, o modelo orientado a objetos é mais natural (mais próximo de nossa compreensão da realidade), e mais econômico no sentido de que necessita de menos passos para atingir um mesmo resultado. (MOREIRA NETO, 2009).

A POO potencializa a reutilização de código através da herança onde uma classe pode ser aprimorada através de uma nova classe estendendo as suas funcionalidades economizando tempo no desenvolvimento de novas aplicações (DEITEL; DEITEL, 2010).

#### 4.5.4 Enterprise JavaBeans

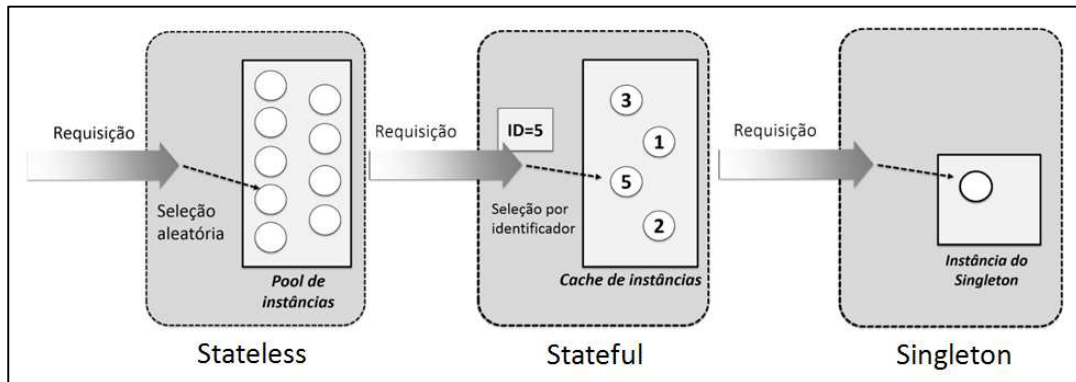
O *Enterprise JavaBeans* (EJB) encapsulam as regras de negócio do sistema e compõem a arquitetura do JEE (ORACLE 2009a, ORACLE 2009b). Esses componentes para funcionarem necessitam de um contêiner que funcione como um hospedeiro de aplicações. O contêiner tem a responsabilidade de fornecer os serviços para as aplicações hospedadas que para habilitar os serviços devem conter as configurações adicionais definidas por XML (*eXtensible Markup Language*) ou anotações (RUBINGER; BURKE, 2010). O contêiner pode prover serviços como controle de transação, concorrência, pool de conexões com banco de dados, gerenciamento do ciclo de vida de objetos segurança entre outros, facilitando assim o desenvolvimento de aplicações já que o desenvolvedor passa a utilizar o serviço e não os implementar (ORACLE 2009a, ORACLE 2009b).

A partir da versão 3.x do EJB, passaram a existir somente dois tipos de componentes: *Session Beans* e *Message Driven Beans*. Na versão anterior a 2.x, existia um terceiro tipo denominado *Entity Bean* responsável por manipular informações com o banco de dados. Este fato ocorreu pela criação do JPA (*Java Persistence API*). Entretanto o *Entity Bean* ainda existe nas especificações mas apenas para manter a compatibilidade (RUBINGER; BURKE, 2010).

Os *Session Beans* são divididos em três tipos: *Stateless*, *Stateful* e *Singleton* (ORACLE 2009b). O *Stateless*, como o próprio nome sugere, é um componente que não mantém estado, ou seja, ele não mantém informações de suas variáveis onde a sua utilização volta para o *pool* para ficar disponível para uma nova solicitação. O *Stateful* faz com que o contêiner tenha que gerenciar a manutenção do seu estado, ou seja, em diversas requisições o servidor irá retornar o mesmo componente com seus devidos atributos mantidos. Quanto ao *Singleton*, o contêiner

mantém somente uma única instância da classe. Na Figura 30 (adaptado de RUBINGER; BURKE, 2010) são representados as diferentes formas de gerenciamento realizado pelo contêiner conforme o tipo de *Session Beans*.

**Figura 30: Gerenciamento das requisições dos Session Beans pelo contêiner**



Fonte: Elaborada pelo autor.

#### 4.5.5 Banco de dados

Um banco de dados possui a finalidade de manter uma coleção de dados, armazenando-os e relacionando-os em arquivos em um servidor (BRYLA; LONEY, 2009). A gerência do acesso e manutenção destes dados se dá através do Sistema de Gerenciamento do Banco de Dados (SGBD) que é utilizado por diversos tipos de aplicações.

Uma das formas de classificar um SGBD é pela maneira que ele organiza e manipula os dados armazenados. Dentre estas formas, as três principais são SGBDR (relacional), SGBDO (objeto) e SGBDOR (objeto-relacional) (LONGLEY et al., 2013).

O SQL é a linguagem utilizada para a manipulação de dados, onde não se trata de uma linguagem procedural com a finalidade de criação de funções ou procedimentos. Os comandos SQL são divididos em três categorias (FREEMAN, 2009):

- Data Control Language (DCL)* - Tem a finalidade de gerenciar o acesso aos dados e estruturas;
- Data Definition Language (DDL)* - Possui comandos para a criação das estruturas para armazenamento e manipulação de dados;
- Data Manipulation Language (DML)* - Utilizados para a manipulação dos dados permitindo a seleção, inclusão, alteração e deleção.

Apesar da linguagem SQL ser a linguagem padrão de consulta a banco de dados, na maioria do desenvolvimento da aplicação o seu uso foi implícito, ou seja, foi abstraído com a utilização de um *framework* para a persistência de dados detalhados na seção a seguir.

#### 4.5.6 Camada de persistência

A linguagem de desenvolvimento utilizada para o projeto foi o Java devido a ser uma poderosa linguagem que permite o desenvolvimento de aplicações *Web*, desktop e para dispositivos móveis.

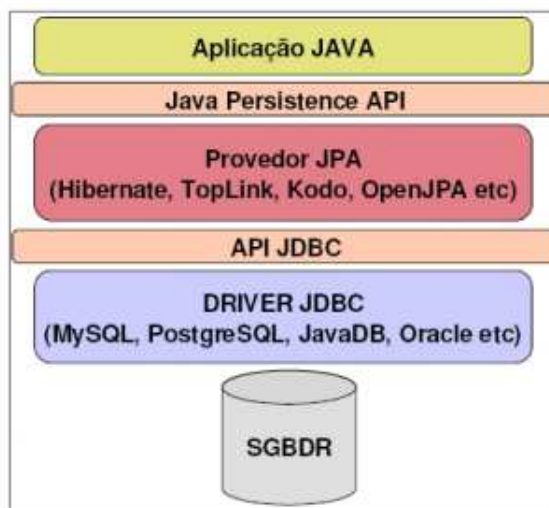
Atualmente, cresce a demanda por aplicações corporativas distribuídas que requerem acesso a banco de dados com velocidade, segurança e confiabilidade. Para atender esta demanda, o Java oferece o JEE (*Java Enterprise Edition*) que oferece um modelo multicamadas distribuído, flexível a transações, com segurança, suporte a serviços *Web* e componentes reutilizáveis utilizando dados integrados de padrões abertos.

O Java define a partir da JSR-220 (*Java Specification Requests*) uma especificação para ORM (*Object Relation Mapping*) denominada JPA (*Java Persistence API*). Atualmente, no mercado existem inúmeras implementações dentre elas é possível destacar: *Hibernate*, *TopLink*, *Kodo*, *OpenJPA* (CORDEIRO, 2011).

Com a utilização de um *framework* de persistência é possível desacoplar a aplicação do banco de dados, ou seja, a aplicação passa a poder, com poucos ajustes, rodar em vários bancos de dados. Isto é possível devido ao uso de driver específico do banco de dados ao qual vai persistir denominado JDBC (*Java Database Connectivity*).

Conforme Lopes (2008), segue a Figura 31 detalhando as camadas da API de persistência.

**Figura 31: Arquitetura do JPA – Java Persistence Api**



**Fonte:** Adaptada de Lopes (2008).

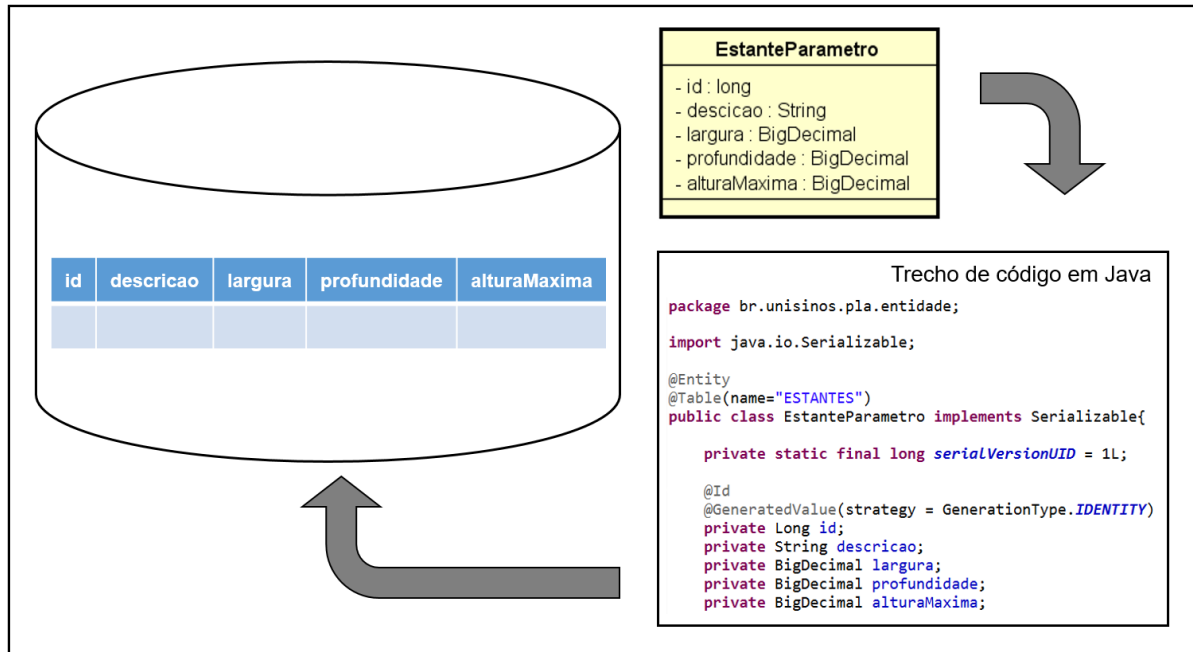
Neste trabalho, foi utilizado o Hibernate devido a estar integrado de forma nativa ao servidor de aplicação e também por se tratar de uma implementação madura compatível com diversos bancos de dados e uma vasta documentação disponível, o que facilitou na curva de aprendizado (CORDEIRO, 2011).

A meta de Hibernate é aliviar o desenvolvedor em 95% de dados comuns de persistência relacionados às tarefas de programação. O Hibernate talvez não seja a melhor solução para as aplicações centradas em dados, das quais apenas usam procedimentos armazenados para a implementação das lógicas comerciais no banco de dados. Isto é mais utilizado orientando o objeto aos modelos de domínio e lógicas comerciais na camada intermediária baseada em Java. (KING et al., 2011).

No mapeamento das entidades de relacionamento com o banco de dados, optou-se por utilizar as anotações disponibilizadas pela biblioteca `javax.persistence`, que definem as propriedades da tabela a partir da classe criada em Java. O modelo de desenvolvimento aplicado foi *top-down*, ou seja, primeiramente foi desenvolvido o código Java e o Banco de dados foi gerado automaticamente ao iniciar a aplicação.

A Figura 32 apresenta um trecho do código exemplificando o fluxo de criação da base de dados.

**Figura 32: Fluxo de criação da base dados**



**Fonte:** Elaborada pelo autor.

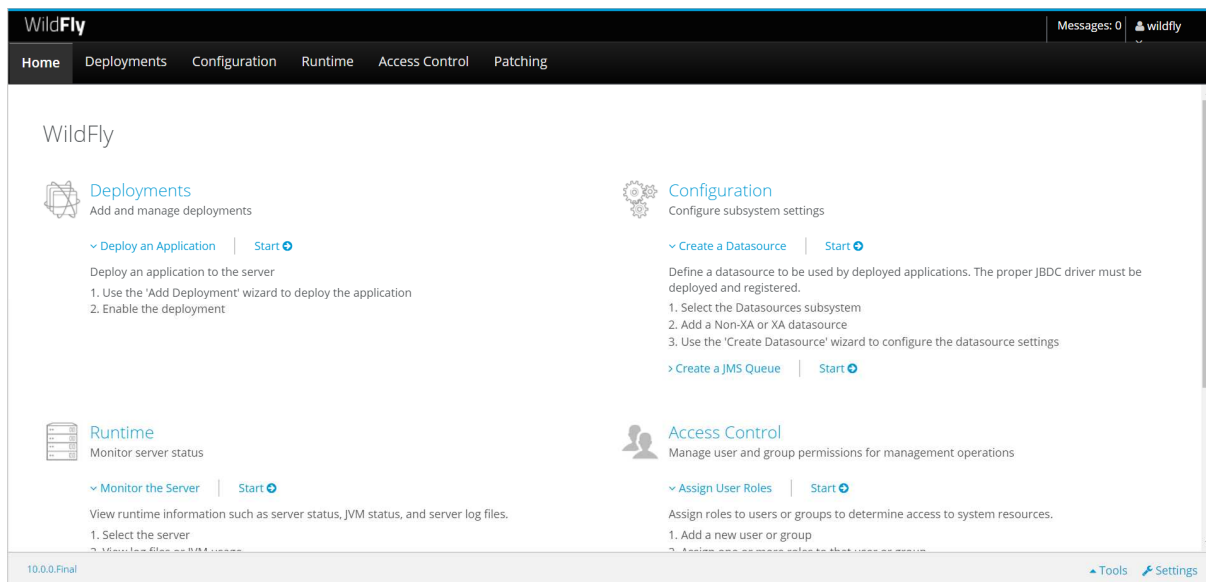
#### 4.5.7 Servidor de Aplicação *Web*

Devido à tecnologia empregada no projeto JEE, o servidor de aplicação *Web* passa a ter a responsabilidade de fornecer um contêiner capaz de comportar as especificações da tecnologia. Dentre as soluções disponíveis no mercado, optou-se pelo Wildfly por oferecer a infraestrutura necessária para o desenvolvimento de aplicações JEE.

É uma solução modular onde é possível ativar e desativar recursos conforme necessidade, disponibiliza as últimas especificações disponíveis pela tecnologia Java. Tem uma inicialização muito rápida e um baixo consumo de memória devido a sua arquitetura que visa ao compartilhamento de recurso entre os seus componentes.

Dentre os recursos disponibilizados, se destaca o gerenciamento e implantação das aplicações desenvolvidas que podem ser realizadas pelo console *Web* conforme é possível verificar na Figura 33.

**Figura 33: Console de administração do servidor Wildfly**



**Fonte:** Elaborada pelo autor.

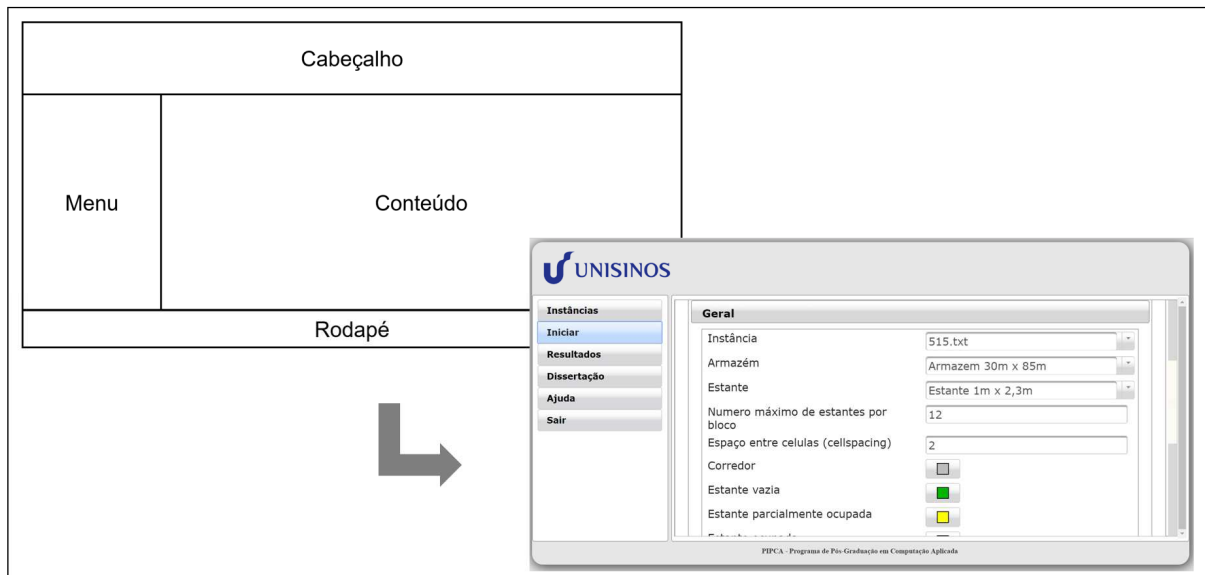
#### 4.5.8 Interface

Para o desenvolvimento da interface, se fez uso de duas tecnologias Java: O JSF (*Java Server Faces*) e o *Facelets*. O uso do *Facelets* otimiza o desenvolvimento, diminuindo o tempo gasto no projeto pois permite a criação de *templates* que podem ser empregados na construção das páginas.



Para o trabalho, foi criado um *template* onde se fez a criação das áreas comuns a todas as páginas, deixando o conteúdo da página para ser detalhado conforme é possível visualizar na Figura 34.

**Figura 34: Template aplicação geral da aplicação**



**Fonte:** Elaborada pelo autor.

O JSF é uma tecnologia do JEE para o desenvolvimento de aplicações *web*. As tecnologias herdadas pelo JSF são o *Java Server Pages (JSP)* e *Servlets*, estendendo os seus conceitos de ciclo de vida e garantindo um conjunto de recursos com foco para o desenvolvimento *web* (GOMES, 2008).

No JSF, o controle é composto por um *servlet* denominado *FacesServlet*, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O *FacesServlet* é responsável por receber requisições da *WEB*, redirecioná-las para o modelo e então remeter uma resposta (PITANGA, 2004).

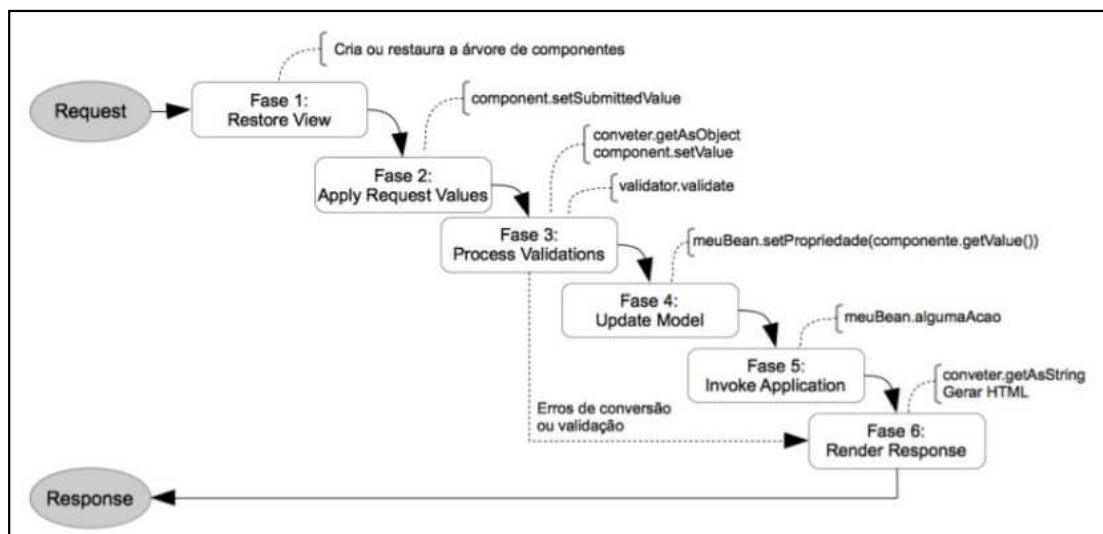
Por se tratar de um framework que utiliza componentes, o desenvolvedor não tem a preocupação com a transformação dos elementos utilizados em código interpretável pelos navegadores.

Os componentes do JSF possuem um ciclo de vida composto por seis fases (CORDEIRO, 2011). É necessário para o desenvolvimento de aplicações que utilizam JSF o conhecimento de seu ciclo de vida. Os ciclos de vida são apresentados na Figura 35 (CORDEIRO, 2011):

- a) Fase 1 - *Restore View*: É a fase responsável pela criação ou restauração da árvore de componentes JSF utilizadas para a construção da página. No caso de não localizar nenhum erro, ele passa para a Fase 2;
- b) Fase 2 - *Apply Request Value*: Nesta etapa insere os atributos em seus componentes, sem a validação dos valores;
- c) Fase 3 - *Validate*: É a etapa onde os valores são convertidos e validados de acordo com os valores informados no Bean;
- d) Fase 4 - *Update Model*: O framework faz a atualização da classe que está interagindo com a tela;
- e) Fase 5 - *Invoke Application*: É a etapa em que os comandos são processados;
- f) Fase 6 - *Render Response*: Tem a finalidade de interagir com o usuário, gerando mensagem de sucesso ou uma mensagem de erro no caso de problemas durante a execução.

Fonte: (CORDEIRO, 2011).

Figura 35: Ciclo de vida do JSF



## 5 TESTES E RESULTADOS

O presente capítulo apresenta os experimentos e calibrações realizadas, as principais contribuições do modelo, bem como os trabalhos futuros para a continuidade desta pesquisa.

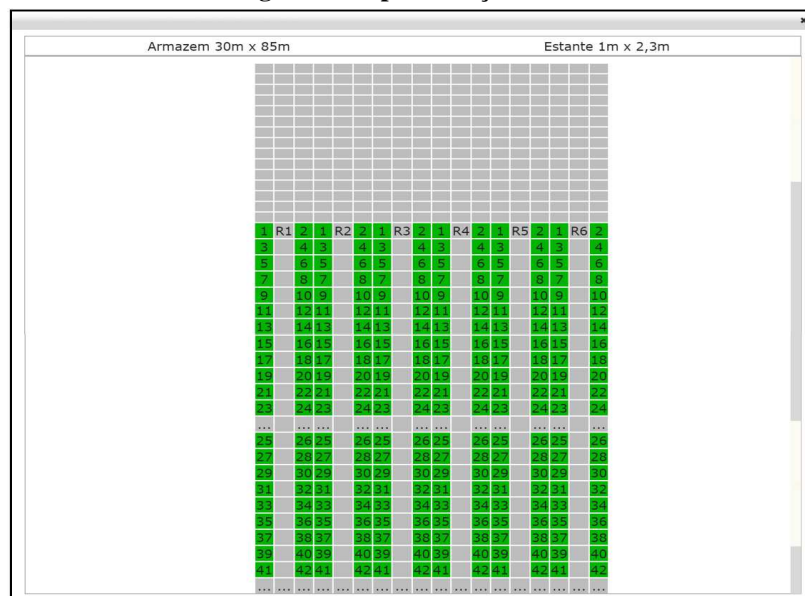
### 5.1 Experimentos e Calibração

Os experimentos descritos nesta seção foram realizados em um computador com processador Intel (R) Core (TM) CPU i7-7500 @ 2,7 GHz 2,9 GHz, 8 GB de memória RAM e sistema operacional Windows 10.

Para a realização dos experimentos, foram geradas três instâncias conforme as características descritas na Tabela 5, apresentada na seção 4.4. A primeira instância gerada possui 100 produtos denominada “i-100”, a segunda 300 produtos, denominada “i-300” enquanto a terceira ficou com 500 produtos, denominada “i-500”.

Além da definição das instâncias, um outro fator bastante importante é a definição do armazém e das estantes. Para este experimento, optou-se pelo uso de um armazém de dimensões de 30 metros de largura com 80 metros de comprimento, enquanto a estante com 1 metro de profundidade com 2,3 metros de largura com altura máxima para endereços de coleta de 2,4 metros. Com as dimensões do armazém e das estantes utilizadas, foram distribuídas 252 estantes. A Figura 36 apresenta o armazém utilizado.

**Figura 36: Apresentação do armazém**



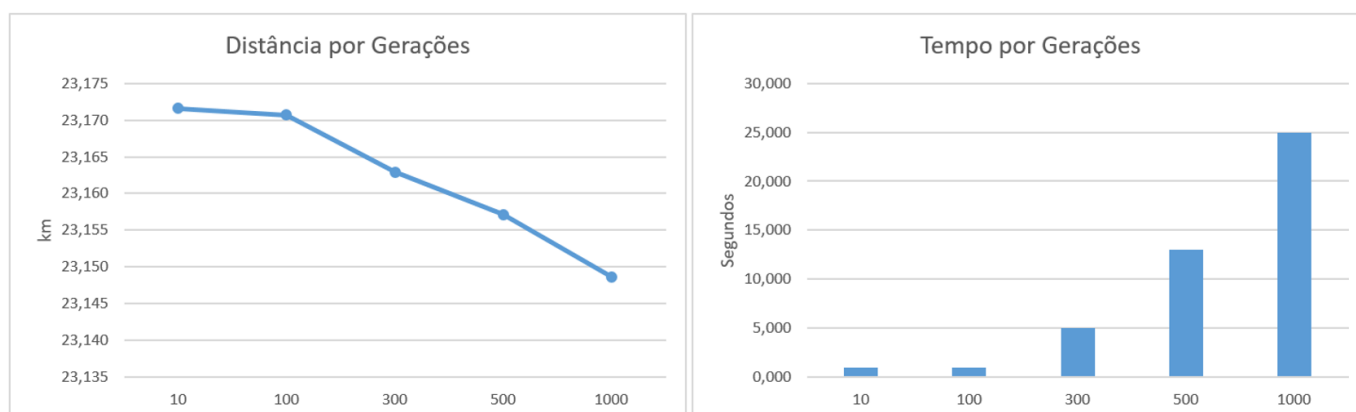
**Fonte:** Elaborada pelo autor.

A calibração é necessária para o ajuste dos parâmetros do AG com a finalidade de conseguir alcançar um custo benefício aceitável de tempo necessário para a execução em relação à solução que será apresentada como o melhor resultado encontrado (WICKERT, 2012).

Dentre os parâmetros que foram testados, estão: O número de gerações, o tamanho da população e a taxa de mortalidade. Para a realização da calibração, utilizou-se a instância “i-100”. Para cada cenário, realizou-se 50 execuções onde o foi analisado a distância média gerada e o tempo médio necessário para o término de cada execução.

O primeiro cenário testado foi o número de gerações, onde neste teste a população foi fixada em 100 indivíduos e taxa de mortalidade em 10%. Foram realizados um total de 150 execuções divididas em 5 definições diferentes, sendo elas: 10, 100, 300, 500 e 1000. Após a realização do experimento, nota-se que a partir de 500 gerações a melhora começa a reduzir na distância encontrada, onde o tempo necessário para cada execução aumenta de acordo com o número de gerações. A Figura 37 apresenta os resultados.

**Figura 37: Número de gerações x tempo e distância**

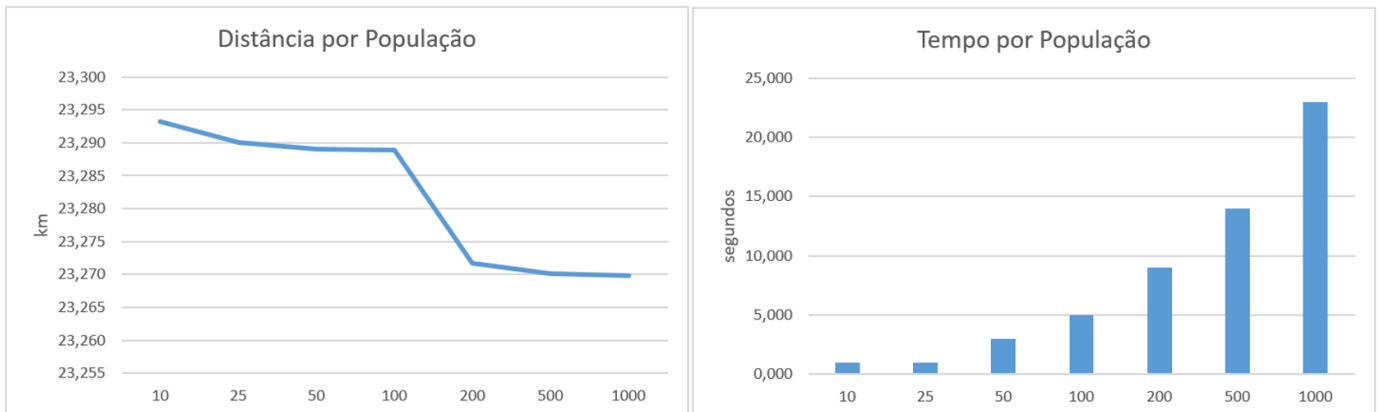


**Fonte:** Elaborada pelo autor.

A Figura 37 apresenta os resultados encontrados após a realização dos experimentos quanto ao número de gerações. É notório que quanto mais gerações são executadas maior é o tempo por execução.

O segundo cenário testado foi o tamanho da população, onde neste teste o número de gerações foi fixado em 10%. Neste experimento foram realizados um total de 210 execuções distribuídas em 7 tamanhos diferente sendo elas: 10, 25, 50, 100, 200, 500 e 1000. A figura 38

**Figura 38: Tamanho da população x distância e tempo**

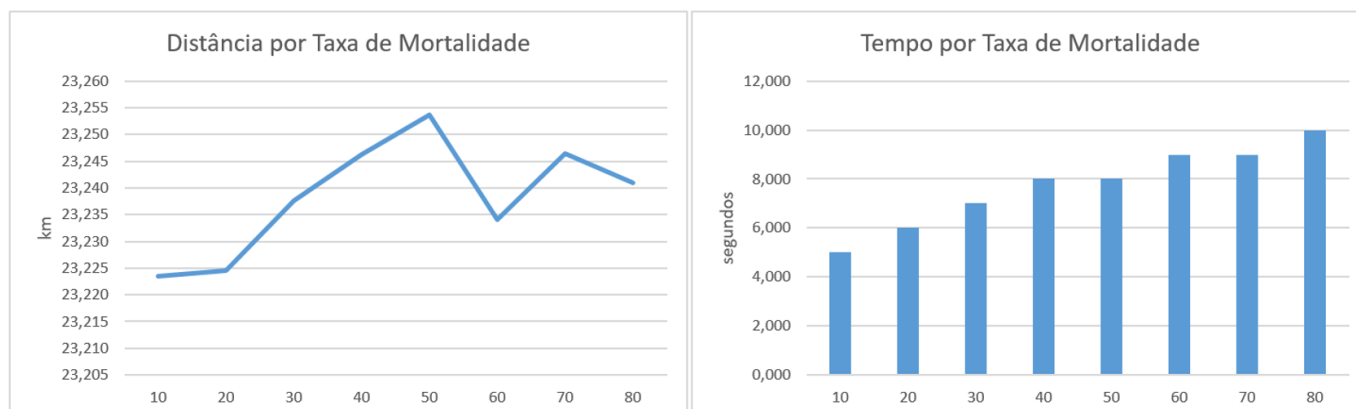


apresenta os resultados obtidos.

**Fonte:** Elaborada pelo autor.

A Figura 38 apresenta os resultados obtidos, sendo notória uma estabilização a partir de uma população com 200 indivíduos. No primeiro cenário, todos os testes foram executados com uma população de 100 indivíduos onde ao analisar o caso mais extremo em ambos, 1000 ciclos contra uma população de 1000 indivíduos o tempo necessário é um pouco menor, isto se deve à característica da solução apresentada, onde quanto maior o número de ciclos mais processos são realizados, enquanto o tamanho da população não necessariamente irá gerar mais processos.

O terceiro experimento de calibração tratou de analisar o impacto e o custo da taxa de mortalidade, onde o tamanho da população e numero de gerações ficou em 100. Neste experimento, foram realizadas 240 execuções divididas em 8 percentuais sendo eles: 10, 20, 30, 40, 50, 60, 70 e 80. A Figura 39 apresenta os resultados obtidos.

**Figura 39: Taxa de mortalidade x distância e tempo**

**Fonte:** Elaborada pelo autor.

A Figura 39 apresenta os resultados encontrados mostrando uma estabilidade com pouca variação entre os percentuais testados. Os percentuais entre 10 e 20 por cento apresentaram um melhor resultado. Conforme era de se esperar, quanto maior a taxa de mortalidade maior o tempo necessário para cada execução, devido ao custo maior para a criação de um novo indivíduo sem a utilização de cruzamento.

Após a etapa de calibração do AG, iniciou-se a realização da coleta dos resultados nas instancias estabelecidas. Foram realizadas 100 execuções considerando como parâmetros: trezentas gerações, tamanho da população com duzentos indivíduos e taxa de mortalidade com 10%. A Tabela 6 apresenta os valores encontrados nos experimentos.

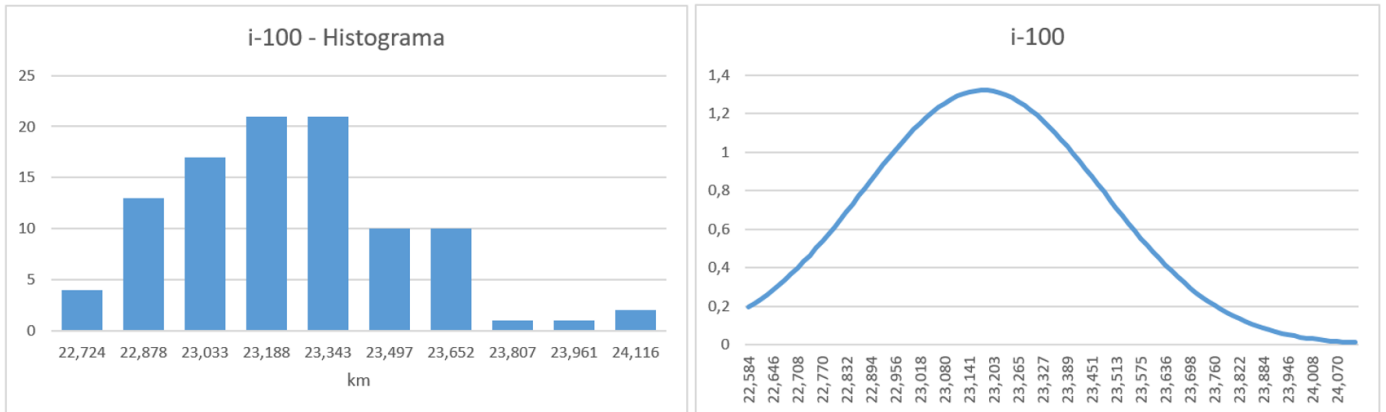
**Tabela 6: Resumo dos resultados**

Instância	Nro execuções	Distância em km			Tempo em segundos	
		Melhor	Pior	Média	Médio	Total
i-100	100	22,569	24,116	23,165	5,45	545
i-300	100	74,092	86,244	79,632	26,85	2685
i-500	100	136,103	156,327	145,717	57,36	5736

**Fonte:** Elaborada pelo autor.

O tempo total para a geração dos resultados foi de 8966 segundos, aproximadamente 150 minutos. Nas instâncias “i-300” e “i-500”, o percentual de melhora entre o pior caso e o melhor caso chegou próximo aos 15% enquanto na menor instância “i-100” ficou acima de 5%. Na Figura 40, segue os resultados detalhados do experimento da instância “i-100”.

**Figura 40: Histograma e distribuição normal da instância i-100**

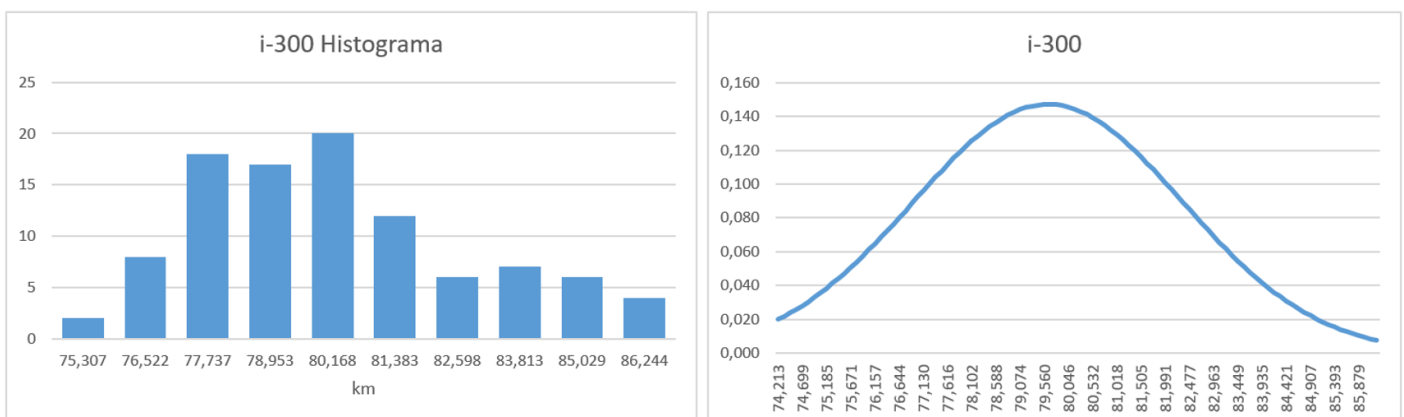


**Fonte:** Elaborada pelo autor.

A Figura 40 apresenta o histograma e a distribuição normal dos resultados encontrados para a instância “i-100”. O resultado com maior frequência foi na faixa de 23,188 a 23,343 com 42 ocorrências. A média encontrada foi de 23,165 com desvio padrão de  $\pm 0,302$  obtendo uma melhora de aproximadamente 1,5 km da melhor solução comparada com a pior solução.

Nos resultados da instância “i-300”, a média encontrada foi de 79,632 com desvio padrão de  $\pm 2,712$ . Quanto à frequência, o resultado de maior expressão foi 80,168 com 20 ocorrências. Neste experimento, o melhor resultado quando comparado com o pior resultado apresentou uma diferença de aproximadamente 12,152 km. Abaixo, segue a Figura 41 apresentando o histograma e a distribuição normal.

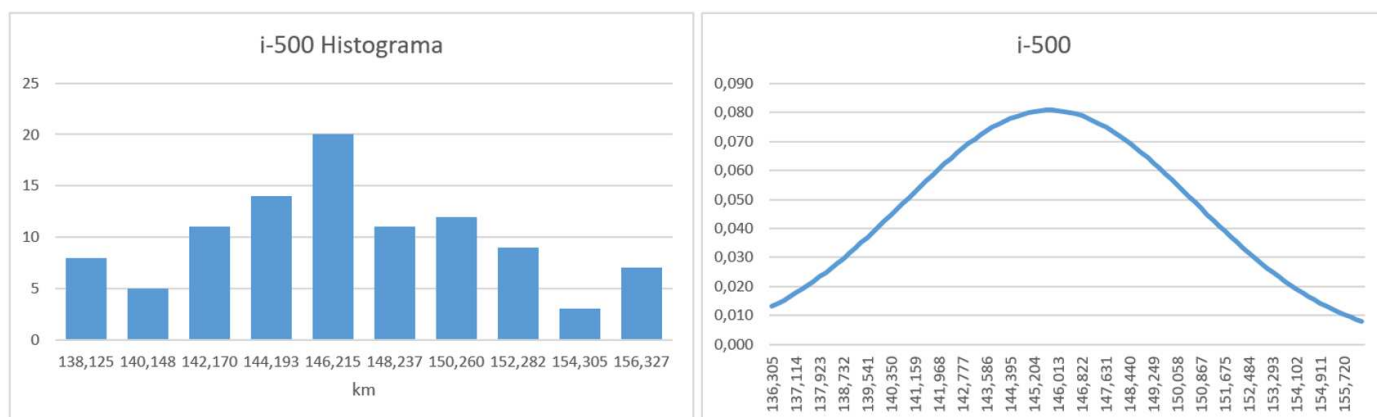
**Figura 41: Histograma e distribuição normal da instância i-300**



**Fonte:** Elaborada pelo autor.

Na instancia “i-500”, a diferença entre o melhor resultado e o pior chegou a mais de 20 km. A média foi de 145,717 com desvio padrão de  $\pm 4,942$ , enquanto o resultado de maior expressão foi na faixa de 146,215 com 20% das ocorrências. Abaixo, segue a Figura 42

**Figura 42: Histograma e distribuição normal da instância i-500**



**Fonte:** Elaborada pelo autor.

## 5.2 Comparação com outros resultados

Conforme apresentado na seção 3.5, existem inúmeros trabalhos que possuem relação direta ou indireta com a pesquisa realizada. Dentre os assuntos relacionados, trabalhos que visam à geração de um layout para um CD ou tenha como objetivo otimizar o processo de seleção de pedidos, de certa forma tem relação com este trabalho.

Dentre os trabalhos analisados, pode-se destacar o trabalho “*Layout design modelling for a real world just-in-time warehouse*” realizado no ano de 2013 com a finalidade de otimizar a operação de seleção de pedidos em um *crossdocking*. Outro trabalho que apesar de não projetar um layout tradicional é o “*Detailed design of fishbone warehouse layouts with vertical travel*” que tinha como principal objetivo a geração de um layout para um centro de distribuição. (CARDONA et al., 2015; HORTA et al., 2016). Abaixo, segue a Tabela 7 apresentando os computadores utilizados nos experimentos.

**Tabela 7: Processador e memória utilizado pelos trabalhos**

Trabalho	Processador	Memória
Uma aplicação de metaheurísticas na abordagem do problema de <i>layout</i> de armazém	Intel (R) Core (TM) CPU i7-7500 2,7 GHz 2,9 GHz	8 GB
<i>Layout design modelling for a real world just-in-time warehouse</i>	Intel Core 2,4 GHz	4 GB
<i>Detailed design of fishbone warehouse layouts wit vertical travel</i>	Intel (R) Core (TM) CPU i7-4500 1,8 GHz 2,4 GHz	8 GB



**Fonte:** Elaborada pelo autor.

Em seu trabalho, Cardona et al. (2015) cria um modelo com a finalidade de gerar um *layout* tridimensional não tradicional, que visa minimizar os custos na operação dentro do CD. Algumas variáveis não estão presentes em seu modelo, onde os produtos armazenados são sacos de 50 kg, ou seja, não existe variação de dimensões e peso.

Por outro lado, Horta et al. (2016), apesar de possuir uma finalidade similar, utiliza um *layout* tradicional e trabalha com produtos que possuem pesos e formas variadas. Dentro do seu estudo, o CD trabalha com verdura e frutas que são produtos perecíveis, que traz uma complexidade maior para a operação onde a estocagem deve ser evitada ao máximo.

Apesar das diferenças entre os trabalhos, seja quanto ao cenário ou ao modelo estabelecido foi criada a Tabela 8 com a finalidade de realizar a comparação entre os casos mais extremos de cada trabalho.

**Tabela 8: Comparação entre os trabalhos**

Caso mais extremo		
Trabalho	Iterações	Tempo
<i>Detailed design of fishbone warehouse layouts wit vertical travel</i>	Não informado	4,5 minutos
<i>Layout design modelling for a real world just-in-time warehouse</i>	5418395	24 horas
Uma aplicação de metaheurísticas na abordagem do problema de <i>layout</i> de armazém	5001000	57,36 minutos

**Fonte:** Elaborada pelo autor.

O trabalho realizado por Cardona et al. (2015) precisou de 4,5 minutos para terminar a sua execução, que percentualmente consumiu somente 7,84% do tempo necessário neste trabalho, entretanto, o mesmo não destaca o número de iterações realizadas. Por outro lado, o número de iterações similares no trabalho realizado por Horta et al. (2016) levou 22,16 vezes mais tempo para alcançar o seu resultado.



## 6 CONCLUSÕES

Durante todo o trabalho, realizou-se estudos com a finalidade de gerar uma solução para problema de *layout* de armazém. Os resultados deste trabalho tratam-se de um *layout* de armazém tradicional, caracterizado pela disposição dos corredores paralelos e ortogonal em relação às paredes (CARON; MARCHET; PEREGO, 2000).

Napolitano (2012) alega que, em torno de 80%, os armazéns realizam suas atividades de forma manual, onde a melhora na disposição das mercadorias impacta diretamente na diminuição do tempo para a seleção dos pedidos, pois reduzem a distância percorrida pelo colaborador durante este processo.

A seleção de pedidos em muitos armazéns representa em torno de 50% dos custos com recursos humanos. Dentro dos processos realizados durante a seleção de pedidos, o percurso representa em torno de 50% do tempo gasto.

Os resultados obtidos neste trabalho são promissores, chegou-se a melhoras de aproximadamente 15% comparando a pior e a melhor solução encontrada quanto a distância para o *layout*. Este custo, conforme descrito na seção 4.3, trata-se do somatório da localização da mercadoria multiplicado pela média de pedidos por dia.

Inúmeros trabalhos são encontrados na literatura para o problema, entretanto, cada um possui modelagem e particularidades que muitas vezes os tornam singulares apesar de tratar do mesmo problema.

Quando analisado o trabalho “*Layout design modelling for a real world just-in-time warehouse*”, que apesar de se tratar de um trabalho similar mas com modelagem e problema diferente do atual, é possível realizar uma comparação indireta onde, apesar de conseguir percentuais próximos de 20% de melhora, o tempo necessário para conseguir a solução foi próximo de 24 horas (HORTA et al., 2016).

Outro trabalho que aborda a criação de layout não convencional denominado “*Detailed design of fishbone warehouse layouts with vertical travel*” também alcança resultados próximos à 20% na maioria dos casos estudados com tempos próximo a 4,5 minutos (CARDONA et al., 2015).

Apesar da maior instância utilizada neste trabalho possuir 500 produtos, o trabalho em questão não tem limites quanto à dimensão do armazém ou ao número de produtos a serem alocados. Neste trabalho o tempo para a instância “i-500” foi de 57,36 minutos.

## 6.1 Trabalhos futuros

Este trabalho estudou um assunto com muita aplicabilidade, onde são muitas as oportunidades de pesquisas que surgem a partir deste trabalho, como:

- a) Criar mecanismo para a otimização da armazenagem;
- b) Expandir o problema para a armazenagem geral das mercadorias, ou seja, não considerar somente os endereços de coleta;
- c) Ampliar a otimização analisando as possíveis rotas mediante a uma carga de pedidos;
- d) Considerar outras atividades dentro do armazém como a armazenagem;

Um ponto importante para a aplicabilidade prática é gerar uma interface para a importação de dados de armazéns já existentes, a fim de apresentar novas disposições para os produtos juntamente com o percentual de melhora.

Uma interface capaz de apresentar o *layout* em três dimensões conseguiria transformar o trabalho realizado em um simulador, a fim de proporcionar à empresa que utilizasse uma amostra elucidativa das possíveis soluções.

## REFERÊNCIAS

ACKERMAN, K. B. **Practical handbook of warehousing**. 4th ed. London: Chapman & Hall, 1997.

AL JADAAN, O.; RAJAMANI, L.; RAO, C. R. Improved Selection Operator for GA. **Journal of Theoretical and Applied Information a Technology**, Osmania, v. 4, n. 4, p. 269-277, 2008.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos/logística empresarial**. 5. ed. Porto Alegre: Bookman, 2006.

\_\_\_\_\_. **Logística Empresarial**: transportes, administração de materiais e distribuição física. São Paulo: Atlas, 1995.

BARROS, M. C. Warehouse management system (WMS): conceitos teóricos e implementação em um centro de distribuição. 2005. 132 f. Dissertação (Mestrado em Logística) – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

BISCHOFF, E. E. Three dimensional packing of items with limited load bearing strength. **European Journal of Operation Research**, v. 168, n. 3, p. 952-966, feb. 2004.

BORGES, D. A. Uma solução integrada aplicada ao problema de otimização do ciclo de montagem de uma insersora automática de componentes utilizando uma abordagem híbrida de metaheurísticas. 2009. 126 f. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2009.

BOWERSOX, D. J. et. al. **Gestão logística da cadeia de suprimentos**. Porto Alegre: Bookman, 2006.

BOWERSOX, D. J.; CLOSS, D. J. **Logística empresarial**: o processo de integração da cadeia de suprimento. São Paulo: Atlas, 2001.

BOWERSOX, D. J; CLOSS, D. J.; COOPER, M. B. **Supply chain logistics management**. 3rd ed. New York: McGraw-Hill, 2010.

BRAGHIROLI, L. F. Estudo da linearidade da produção em células de manufatura através de simulação a eventos discretos. 2009. 182 f. Dissertação (Mestrado em Sistema de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2009.

BRITO, E. S. Aplicativo para modelamento 3D de layout celular com base em tecnologia de grupo. 2010. 97 f. Dissertação. (Mestrado em engenharia) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.

BRUM, J. G. F. Desenvolvimento de um protótipo de *software* para geração de grade de programação de comerciais aplicável à TV Digital/IPTV utilizando Metaheurísticas. 2014. 169 f. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2014.

- BRYLA, B.; LONEY, K. **Oracle Database 11g Manual do DBA**. Porto Alegre: Bookman, 2009.
- CARDONA, L. F. et al. Detailed design of fishbone warehouse layouts with vertical travel. *International Journal of Production Economics*, v. 170, Part C, p. 825-837, dec. 2015.
- CARON, F.; MARCHET, G.; PEREGO, A. Layout design in manual picking systems: a simulation approach. *Integrated Manufacturing Systems*, v. 11, n. 2, p. 94–104, 2000.
- CARVALHO, J. C. et. al. **Logística e gestão da cadeia de abastecimento**. Lisboa: Edições Silabo, 2012.
- CHING, Y. H. **Gestão de estoques na cadeia de logística integrada: supllly chain**. 3. ed. São Paulo: Atlas, 2007.
- CHIWIACOWSKY, LD.; MONTEIRO R dos S. Uma aplicação da metaheurística algoritmo genético ao problema de carregamento de navios-contêineres Rio de Janeiro: **PODes - Revista Eletrônica Pesquisa Operacional para o Desenvolvimento**, v.4, n.1, p. 52-64, 2012.
- CHU, C. H. Clustering analysis in manufacturing cellular formation. **Omega: International Journal of Management Sciences**, v. 17, n. 3, p. 289-295, 1989.
- CORDEIRO, J. S. T. Estudo comparativo entre os frameworks de mapeamento objeto-relacional hibernate e toplink. 2011. 55 f. Trabalho de Conclusão de Curso (Especialização) – Departamento de Informática Especialização em Desenvolvimento de Sistemas para *Web*, Universidade Estadual de Maringá Centro de Tecnologia, Maringá, PR, 2011.
- CUNHA, C. B, et al. Heurísticas para o problema de bin-packing no contexto da distribuição física de produtos. In: XL SBPO – 40. Simpósio Brasileiro de Pesquisa Operacional, 2008, João Pessoa. **Anais...** (CD-ROM), p. 712-723.
- CZARKOWSKI, D. Identification and Optimization PID parameters using MATLAB. 2002. 67 f. Cork Institute of Technology, Gdynia Maritime University. Poland, 2002.
- DALMAS, V. Avaliação de um layout celular implementando: um estudo de caso em uma indústria de autopeças. 2004. 133 f. Dissertação (Mestrado em Engenharia) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.
- DEB, K. **Multi-objective optimization using evolutionary algorithms**. New York: John Wiley Sons, 2001.
- DEITEL, P. J.; DEITEL, H. M. **Java: como programar**. 8. ed. São Paulo: Pearson Education do Brasil, 2010.
- DIAS, C. R. Algoritmos evolutivos para o problema de clusterização de grafos orientados: desenvolvimento e análise experimental. 2004. 129 f. Dissertação (Mestrado) – Universidade Federal Fluminense (UFF), Niterói, 2004.

DUKIC, G.; CESNIK, V.; OPETUK, T. Order-picking Methods and Technologies for Greener Warehousing. **Strojarstvo. Journal for Theory and Application in Mechanical Engineering**, v. 52, n. 1, p. 23-31, p. 2010.

EIBEN, A. E.; SMITH, J. E. **Introduction to evolutionary computing**. Berlin: Springer, 2003. (Natural Computing Series).

FERNANDES, A. M. R. **Inteligência artificial: noções gerais**. Florianópolis: Visual Books, 2005.

FLEURY, P. F.; WANKE, P.; FIGUEIREDO, K. F. (Org.) **Logística Empresarial: a perspectiva brasileira**. São Paulo: Atlas, 2000.

FREEMAN, R. G. et al. **OCP: Oracle Database 11g Administrator Certified Professional Study Guide**. Indianapolis: Wiley, 2009.

GALAFASSI, C. Aplicação de metaheurísticas na abordagem do problema de roteamento de veículos capacitado com janela de tempo. 2011. 215 f. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2011.

GASNIER, D.; BANZATO, E. Armazém inteligente. **Revista LOG Movimentação e Armazém**, São Paulo, n. 128, jun. 2001.

GEHRING, H.; BORTFELDT, A. A genetic algorithm for solving the container loading problem. **International transactions in Operation Research**, v. 4, n. 5-64, p. 401-418, nov. 1997.

GLOVER, F. W.; KOCHENBERGER, G. A. (Eds.). **Handbook of metaheuristics**. Boston: Kluwer Academic Publishers, 2003. (International Series in Operations Research & Management Science, v. 57).

GOMES, Y. M. P. **Java na Web com JSF: Spring, Hibernate e Netbeans 6**. Rio de Janeiro: Ciência Moderna, 2008.

GORNI, D. et al. Open source web GIS sistema de informação geográfica de expedições. In: SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, 13., 2007, Florianópolis. **Anais...** Florianópolis: INPE, 2007. p. 3953-3960. Disponível em: <<http://marte.dpi.inpe.br/col/dpi.inpe.br/sbsr@80/2006/11.13.10.51/doc/3953-3960.pdf>>. Acesso em: 15 jul. 2017.

GROOVER, M. P. **Automação industrial e sistemas de manufatura**. 3. ed. São Paulo: Pearson, 2012.

GUIMARÃES, A. O. Algoritmo genético aplicado no controle de posição do rotor de um motor de corrente contínua com rejeição a distúrbios por ação feedforward. 2013. 72 f. Dissertação (Mestrado em Sistemas de Comunicação e Automação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Pós-Graduação, Mossoró, 2013.

HARFF, M. Aplicação de metaheurísticas no desenvolvimento de um modelo de otimização para o processo de codificação de áudio do sistema brasileiro de televisão digital. 2013. 192 f.

Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Rio dos Sinos, São Leopoldo, 2013.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.** Michigan: University of Michigan Press, 1975.

HORSTMANN, C. S.; CORNELL, G. **Core Java: Volume I - Fundamentos.** 8. ed. São Paulo: Pearson Prentice Hall, 2010.

HORTA, M.; COELHO, F.; RELVAS, S. Layout design modelling for a real world just-in-time warehouse. **Computers & Industrial Engineering**, v. 101, p. 1-9, nov. 2016.

HUGOS, M. H. **Essentials of supply chain management.** 2nd ed. New Jersey: Wiley, 2006.

JEON, G.; LEEP, H. R. Forming part families by using genetic algorithm and designing machine cells under demand changes. **Computers & Operations Research**, v. 33, n. 1, p. 263-283, jan. 2006.

KAMINSKI, L. A. Proposta de uma sistemática de avaliação dos custos logísticos da distribuição física: o caso de uma distribuidora de suprimentos industriais. 2004. 130 f. Dissertação (Mestrado em Engenharia – Ênfase em Logística) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2004. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/4747/000459493.pdf?sequence=1>>. Acesso em: 15 jul. 2017.

KING, G. et al. Documentação de referência Hibernate: 3.5.6-Final. 2011. Disponível em: <<https://docs.jboss.org/hibernate/orm/3.5/reference/pt-BR/html/preface.html>>. Acesso em: 15 jul. 2017.

KOSTER, R.; LE-DUC, T.; ROODBERGEN, K. J. Design and control of warehouse order picking: A literature review. **European Journal of Operations Research**, v. 182, n. 2, p. 481-501, oct. 2007.

LAI, K. K., XUE, J., ZHANG, G. Layout design for a paper reel warehouse: A two-stage heuristic approach. **International Journal of Production Economics**, v. 75, n. 3, p. 231-243, feb. 2002.

LAMBERT, D. M.; STOCK, J. R.; VANTINE, J. G. **Administração Estratégica da Logística.** São Paulo: Vantine Consultoria, 1998.

LIMA, M. Armazenagem: considerações sobre a atividade de Picking. In: FIGUEIREDO, K. F.; FLEURY, P. F.; WANKE, P. (Orgs.). **Logística e gerenciamento da cadeia de suprimentos: planejamento do fluxo de produtos e recursos.** São Paulo: Atlas, 2006. (Coleção COPPEAD de Administração).

LINCK, I. C. C. Um Sistema de Codificação de Vídeo para TV Digital - SBTVD. 2012. 118 f. Dissertação (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2012.



- LINDEN, R. **Algoritmos genéticos: uma importante ferramenta da inteligência computacional**. 2. ed. Rio de Janeiro: Brasport, 2008.
- LONGLEY, P. A. et al. **Sistemas e ciência da informação geográfica**. 3. ed. Porto Alegre: Bookman, 2013.
- LOPES, L. H. C. Sistema web para gestão de pautas e atas de reuniões. 2008. 57 f. Trabalho de Conclusão de Curso (Especialização em Informática Empresarial) – Faculdade de Engenharia de Guaratinguetá, Universidade Estadual Paulista, 2008.
- MELO, A. C. **Desenvolvendo aplicações com UML 2.2: do conceitual à implementação**. 3. ed. Rio de Janeiro: Brasport, 2010.
- MOREIRA NETO, O. Criando classe de objetos a partir de um modelo. In: \_\_\_\_\_. **Entendendo e dominando o Java**. 3. ed. São Paulo: Digerati Books, 2009.
- MOURA, R. A. Logística, movimentação e armazenagem de materiais. **Guia do visitante da MOVIMAT**, ano 23, n. 143, p. 6, set. 2002.
- MURATA, T.; ISHIBUCHI, H. Performance Evolution of Genetic Algorithms for Flowshop Scheduling Problems. In: **World Congress on Computational Intelligence**. Orlando: IEEE. 1994. p. 812-817.
- NAPOLITANO, M. 2012 warehouse/DC operations survey: Mixed signals. **Modern Materials Handling**, v. 51, n. 11, p. 48-56, 2012.
- NASCIMENTO, F. T. Algoritmos genéticos para síntese de filtros aplicados em controle por modo deslizante. 2010. Dissertação (Mestrado) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2010.
- NEUMANN, W. P.; DUL, J. Human factors: spanning the gap between OM and HRM. **International Journal of Operations & Production Management**, v. 30, n. 9, p. 923-950, 2010.
- OLIVEIRA, S.; PALETTA, M. A. Tecnologia da Informação aplicada na logística. **Inteligência em Gestão Logística**, n. 17. 2010.
- ORACLE. JSR 316: Java™ Platform, Enterprise Edition 6 (Java EE 6) Specification. In: Java Community Process, 2009a. Disponível em: <<https://jcp.org/aboutJava/communityprocess/final/jsr316/index.html>>. Acesso em: 15 jul. 2017.
- \_\_\_\_\_. JSR 318: Enterprise JavaBeans™, Version 3.1 EJB Core Constructs and Requirements. In: Java Community Process, 2009b. Disponível em: <<https://jcp.org/aboutJava/communityprocess/final/jsr318/index.html>>. Acesso em: 15 jul. 2017.
- PACHECO, M. A. **Algoritmos genéticos: princípios e aplicações**. Rio de Janeiro: ICA-Rio, 1999.

- PALETTA, M. A; SILVA, A. G. Otimizando o layout do armazém através da movimentação eficiente de materiais. Material Institucional da Faculdade de Tecnologia Prof. Luiz Rosa e Centro Universitário Padre Anchieta, 2003. 15 p.
- PAOLESCHI, B. **Logística industrial integrada: do planejamento, produção, custo e qualidade à satisfação do cliente.** 2. ed. São Paulo: Érica, 2009.
- PITANGA, T. JavaServer Faces: a mais nova tecnologia para desenvolvimento web. In: Grupo de Usuários Java, 2004. Disponível em: <[http://www.paulojose.pro.br/Aulas/Tutorial\\_inicial.pdf](http://www.paulojose.pro.br/Aulas/Tutorial_inicial.pdf)>. Acesso em: 15 jul. 2017.
- POCINHO, G. F. C. Análise e melhoria do processo de order-picking num sistema produtivo: caso de estudo. 2013. 103 f. Dissertação (Mestrado em Engenharia e Gestão Industrial) – Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Lisboa, 2013.
- POLTOSI, R. P. Elaboração de escalas de trabalho de técnicos de enfermagem com busca tabu e algoritmos genéticos. 2007. 145 f. Dissertação. (Mestrado em Computação Aplicada) – Universidade do Vale do Rio dos Sinos, São Leopoldo, 2007.
- PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional.** 7. ed. Porto Alegre: McGraw-Hill, 2011.
- QUEIROZ, T. A. et al. Algoritmos para os problemas da mochila e do corte de estoque tridimensional guilhotinado. In: XLI Simpósio Brasileiro de Pesquisa Operacional, 2009, Porto Seguro - BA. XLI SBPO. p. 1-12, 2009.
- REEVES, C. Genetic algorithms. In: GLOVER, F.; KOCHEMBERGER, G. A. **Handbook of Metaheuristics.** Boston: Kluwer, 2003.
- RENNER, G.; EKART, A. Genetic algorithms in computer-aided design. **Computer aided Design**, v. 35, n. 8, p. 709-726, jul. 2003.
- RIBEIRO, P. C. C.; SILVA, L. A. F.; BENVENUTO, S. R. S. O uso de tecnologia da informação em serviços de armazenagem. **Produção**, v. 16, n. 3, p. 526-537, set./dez. 2006.
- RICHARDS, G. **Warehouse Management: A complete guide to improving efficiency and minimizing costs in the modern warehouse.** London: Kogan Page, 2011.
- RUBINGER, A. L.; BURKE, B. **Enterprise JavaBeans 3.1.** 6th ed. Sebastopol, CA: O'Reilly Media, 2010.
- SANTOS, J. N. Evolução Logística no Brasil. In: Portal da Logística, 2010. Disponível em: <<http://www.portaldalogistica.com/2010/02/evolucao-logistica-no-brasil.html>>. Acesso em: 12 ago. 2014.
- SANTOS, R. **Introdução à Programação Orientada a Objetos usando JAVA.** São Paulo: Elsevier, 2003.
- SELLITTO, M. et al. Avaliação de duas alternativas para distribuição de autopeças considerando centros de distribuição - CD's. INGEPRO - Inovação, Gestão e Produção, v. 1, n.1, 2009.

- SEMAAN, G. S. Algoritmos Heurísticos para o Problema de Particionamento de Grafos com Restrições de Capacidade e Conexidade. 2010. 100 f. Dissertação (Mestrado) – Universidade Federal Fluminense (UFF), Niterói, 2010.
- SILVA, R. M. Impactos da utilização de tecnologias logísticas na atividade de separação de pedidos no ambiente de armazenamento de produtos prontos: estudo de caso em uma empresa de serviços. 2010. 138 f. Dissertação (Mestrado em Engenharia de Produções) - Programa de Pós-Graduação em Engenharia da Produção e Sistemas, Universidade do Vale do Rio dos Sinos, São Leopoldo, RS, 2010.
- SOMMERVILLE, I. **Engenharia de software**. 9. ed. São Paulo: Pearson Education do Brasil, 2011.
- SOUZA, M. F. Evolução da Logística. In: Portal da Logística, 15 abr. 2012. Disponível em: <<http://www.portaldalogistica.com.br/2012/04/evolucao-da-logistica.html>>. Acesso em: 15 jul. 2017.
- TALBI, E. G. **Metaheuristics: from design to implementation**. Hoboken, NJ: John Wiley & Sons, 2009.
- TOMPKINS, J. A. et al. **Facilities planning**. 3rd ed. Hoboken, NJ: John Wiley & Sons; 2003.
- \_\_\_\_\_. et al. **Facilities planning**. 2nd ed. New York: John Wiley & Sons, 1996.
- WICKERT, T. I. Um sistema para sugestão e otimização de conteúdo aplicado ao servidor multimídia SBTVD. 2012. 152 f. Dissertação (Mestrado) – Universidade do Vale do Rio Dos Sinos, São Leopoldo, 2012.
- WURMAN, P. R.; D’ANDREA, R.; MOUNTZ, M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, California, v. 29, n. 1, out. 2008. Disponível em: <<https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/2082/1981>>. Acesso em: 15 jul. 2017.
- YALCINOZ, Y.; ALTUN, H. Environmentally constrained economic dispatch via a genetic algorithm with arithmetic crossover. *IEEE AFRICON Conference*, p. 923-928, out. 2002.