

UNIVERSIDADE FEDERAL DO PARANÁ
JAISON VIEIRA

FRAMEWORK PARA A APLICAÇÃO DE ALGORITMOS GENÉTICOS À SISTEMAS
AVANÇADOS DE PLANEJAMENTO E PROGRAMAÇÃO DA PRODUÇÃO (APS)

CURITIBA
2023

JAISON VIEIRA

FRAMEWORK PARA A APLICAÇÃO DE ALGORITMOS GENÉTICOS À SISTEMAS
AVANÇADOS DE PLANEJAMENTO E PROGRAMAÇÃO DA PRODUÇÃO (APS)

Dissertação apresentada como resultado parcial à
obtenção do grau de Mestre em Engenharia de
Manufatura. Curso de Pós-Graduação em Engenharia
de Manufatura, Setor de Tecnologia, Universidade
Federal do Paraná.

Orientador: Prof. Dr. Fernando Deschamps
Coorientador: Prof. Dr. Alessandro Marques

CURITIBA
2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Vieira, Jaison

Framework para a aplicação de algoritmos genéticos a sistemas avançados de planejamento e programação da produção (APS) / Jaison Vieira. – Curitiba, 2023.

1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia de Manufatura.

Orientador: Fernando Deschamps
Coorientador: Alessandro Marques

1. Planejamento da produção. 2. Algoritmos genéticos. 3. Automação industrial. 4. Planejamento dos recursos de manufatura. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia de Manufatura. III. Deschamps, Fernando. IV. Marques, Alessandro. V. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894



MINISTÉRIO DA EDUCAÇÃO
SETOR DE TECNOLOGIA
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA DE
MANUFATURA - 40001016171P2

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA DE MANUFATURA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **JAISON VIEIRA** intitulada: **Framework para a aplicação de algoritmos genéticos a sistemas avançados de planejamento e programação da produção (APS)**, sob orientação do Prof. Dr. FERNANDO DESCHAMPS, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 30 de Março de 2023.

Assinatura Eletrônica
09/04/2023 19:30:41.0
FERNANDO DESCHAMPS
Presidente da Banca Examinadora

Assinatura Eletrônica
26/04/2023 13:18:39.0
CARLOS ALBERTO FADUL CORRÊA ALVES
Avaliador Externo (FUNDAÇÃO CERTI)

Assinatura Eletrônica
11/04/2023 11:12:10.0
EDUARDO ALVES PORTELA SANTOS
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ -
DEPARTAMENTO DE ADMINISTRAÇÃO GERAL E APLICADA)

Assinatura Eletrônica
06/04/2023 14:44:32.0
MARCIO FONTANA CATAPAN
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
12/04/2023 11:27:00.0
ALESSANDRO MARQUES
Coorientador(a) (DEPARTAMENTO DE ENG. MECÂNICA UFPR)

Avenida Coronel Francisco Heráclito dos Santos, 100 - Curitiba - Paraná - Brasil
CEP 81531-980 - Tel: (41) 3361-3123 - E-mail: ufprpgem@gmail.com

Documento assinado eletronicamente de acordo com o disposto na legislação federal Decreto 8539 de 08 de outubro de 2015.

Gerado e autenticado pelo SIGA-UFPR, com a seguinte identificação única: 273384

Para autenticar este documento/assinatura, acesse <https://www.prppg.ufpr.br/siga/visitante/autenticacaoassinaturas.jsp>
e insira o código 273384

*Dedico esta dissertação aos meus pais,
mentores e amigos, pela inestimável inspiração e
apoio em minha jornada acadêmica.*

AGRADECIMENTOS

A minha jornada no mundo do conhecimento não teria sido possível sem o apoio, incentivo e amor das pessoas que fazem parte da minha vida.

Agradeço a Deus por Sua guia e bênçãos nesta jornada, pois sem Ele, nada disso seria possível. Sua presença constante fortaleceu meu espírito e iluminou cada passo do caminho. Sou grato por Sua sabedoria e amor incondicional.

Dedico este trabalho aos meus pais, que me ensinaram a importância da dedicação e perseverança. Vocês foram a luz que iluminou o caminho nas horas mais difíceis, e sem o seu apoio, nada disso teria sido possível.

Agradeço de coração a todos que contribuíram para a realização deste trabalho e fizeram parte do meu crescimento acadêmico e pessoal.

Ao Prof. Dr. Fernando Deschamps, minha gratidão pelo apoio e orientação em todo o projeto, sua expertise foi fundamental para o sucesso desta pesquisa.

A Gabriel Pardini, meu sincero agradecimento pelo apoio na elaboração do algoritmo NSGA-II, sua contribuição foi essencial para alcançarmos resultados significativos.

À Barbara Laura Cidral, meu agradecimento pelo apoio no design da solução OPTimiz.AI, sua visão criativa trouxe uma perspectiva única para o trabalho.

A Jonatha Carlos Carneiro, minha gratidão pelo apoio no desenvolvimento do front-end, sua dedicação ajudou a tornar a interface do projeto mais eficiente e amigável.

A Françoar Santiago da Silva, meu agradecimento pelo apoio na definição da arquitetura, sua expertise técnica foi fundamental para o desenvolvimento do projeto.

À Produza, meu agradecimento por ceder dados e informações para o case real, sua colaboração enriqueceu o estudo e possibilitou uma aplicação prática dos resultados.

Por fim, dedico esta tese a todas as mentes curiosas e inquietas que buscam incessantemente ampliar o horizonte do conhecimento. Que este trabalho possa contribuir de alguma forma para o avanço da ciência e da sociedade como um todo.

Que esta dedicação reflita o meu profundo reconhecimento a cada um que esteve ao meu lado, seja em palavras de incentivo ou em silêncios que demonstraram apoio. A todos vocês, meu mais sincero e emocionado obrigado.

“Se você quer ser bem-sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si.”

Ayrton Senna

RESUMO

O planejamento e a programação são funções essenciais para que os sistemas industriais operem de maneira eficaz, visando atender às demandas de produção dentro do prazo estabelecido. Diversos estudos na literatura propõem modelos matemáticos de Sistemas Avançados de Planejamento e Programação da Produção (APS) para diferentes cenários, levando em conta configurações variadas de recursos e limitações. No entanto, somente recentemente os APS, que utilizam esses modelos, têm sido amplamente discutidos e utilizados, mas ainda não exploraram completamente os conceitos de digitalização e Indústria 4.0. Este trabalho tem como objetivo desenvolver um framework de um Sistemas Avançados de Planejamento e Programação da Produção para o ambiente de manufatura, utilizando sistemas computacionais de inteligência artificial e aplicando conceitos da Indústria 4.0. Inicialmente, é apresentada uma revisão sistemática da literatura realizada por meio da aplicação do método ProKnow-C em sistemas APS. Em seguida, é mostrado o desenvolvimento de um MVP (Minimum Viable Product) de um sistema de APS por meio da abordagem Design Science Research (DSR), buscando gerar mais flexibilidade e autonomia para estes sistemas, dentro dos conceitos de Indústria 4.0. A abordagem DSR foi explorada em suas etapas, resultando no desenvolvimento de um framework para um sistema comercial de APS, com a aplicação do Algoritmo Genético de Classificação Não Dominada II (NSGA-II). São apresentadas as definições das funções objetivas, um estudo sobre os principais sistemas de APS disponíveis no mercado, a estruturação do sistema, o detalhamento do algoritmo no back-end e os resultados obtidos na simulação por meio do desenvolvimento de protótipos e front-end. Com a utilização da metodologia DSR e a aplicação do algoritmo NSGA-II, busca-se contribuir para o desenvolvimento de sistemas de APS mais eficazes e inovadores, que atendam às demandas da Indústria 4.0.

Palavras-chave: sistemas de planejamento avançado de produção (APS); design science research; indústria 4.0; NSGA-II; algoritmo genético.

ABSTRACT

Planning and scheduling are essential functions for industrial systems to operate effectively, aiming to meet production demands within established deadlines. Several studies in the literature propose mathematical models of Advanced Planning and Scheduling (APS) Systems for different scenarios, taking into account various configurations of resources and limitations. However, only recently have APS systems using these models been widely discussed and used, but they have not fully explored the concepts of digitization and Industry 4.0. This work aims to develop a framework for an Advanced Planning and Scheduling System for the manufacturing environment, using artificial intelligence computational systems and applying concepts of Industry 4.0. Initially, a systematic literature review is presented through the ProKnow-C method applied to APS systems. Then, the development of an MVP (Minimum Viable Product) of an APS system is shown through the Design Science Research (DSR) approach, seeking to generate more flexibility and autonomy for these systems within the Industry 4.0 concepts. The DSR approach was explored in its stages, resulting in the development of a framework for a commercial APS system, with the application of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). Objective function definitions, a study on the main APS systems available in the market, system structuring, algorithm detailing in the back-end, and results obtained in the simulation through the development of prototypes and front-end are presented. By using the DSR methodology and applying the NSGA-II algorithm, we aim to contribute to the development of more effective and innovative APS systems that meet the demands of Industry 4.0.

Keywords: advanced planning and scheduling systems (APS); design science research. industry 4.0; NSGA-II; genetic algorithm.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – O ESQUEMA DE MRP TRADICIONAL E AGENDAMENTO.	20
FIGURA 2 – PRODUTO TECNOLÓGICO.	24
FIGURA 3 – ESTRUTURA DE PRODUTO COM CINCO NÍVEIS.	28
FIGURA 4 – EXEMPLO SIMPLES DE UMA ESTRUTURA DE PRODUTO.	31
FIGURA 5 – EXEMPLO DE UMA ESTRUTURA DE PRODUTO.	32
FIGURA 6 – DIAGRAMA ESQUEMÁTICO DO PROBLEMA APS.	33
FIGURA 7 – DIAGRAMA DE UMA ARQUITETURA BÁSICA DE APS.	34
FIGURA 8 – ESTRUTURA DE UM APS.	35
FIGURA 9 – SUBCAMPOS DA INTELIGÊNCIA ARTIFICIAL.	41
FIGURA 10 - FLUXOGRAMA GENERALIZADO DE UM ALGORITMO GENÉTICO.	47
FIGURA 11 – REPRESENTAÇÃO DO CROMOSSOMO.	48
FIGURA 12 – GERAÇÃO DE SOLUÇÃO VIÁVEL.	50
FIGURA 13 – PROCEDIMENTO DE AVALIAÇÃO DE APTIDÃO.	51
FIGURA 14 – ILUSTRAÇÃO DE Crossover BASEADO EM PEDIDO.	52
FIGURA 15 – PROCEDIMENTO GERAL DA ABORDAGEM BASEADA EM AG.	54
FIGURA 16 – FRONTEIRA DE PARETO, COM EXTREMIDADES DEFINIDAS POR YA E YB, ONDE AS MÍNIMAS PARA CADA FUNÇÃO SÃO OBTIDAS.	55
FIGURA 17 – CICLO DAS ETAPAS DA PESQUISA EM DESIGN SCIENCE.	58
FIGURA 18 – REVISÃO SISTEMÁTICA DA LITERATURA.	61
FIGURA 19 – MOCKUP DO PRODUTO TECNOLÓGICO.	67
FIGURA 20 – LOGO OPTIMIZIA.	68
FIGURA 21 – ARQUITETURA DO SISTEMA OPTIMIZIA.	71
FIGURA 22 – FLUXO DO ALGORITMO OPTIMIZIA.	77
FIGURA 23 – BRIEFING DO PROJETO PARA O TIME DE DESIGN.	81
FIGURA 24 – ENTREVISTA COM USUÁRIOS.	82
FIGURA 25 – PROPOSTA DE VALOR DA SOLUÇÃO.	82
FIGURA 26 – COMPETIDORES DE SOLUÇÕES DE APS.	83

FIGURA 27 – REFERENCIAS IDENTIDADE VISUAL PARA A SOLUÇÃO.....	84
FIGURA 28 – REFERÊNCIAS DE <i>DASHBOARDS</i> PARA A SOLUÇÃO.	84
FIGURA 29 – REFERENCIAS DE GRÁFICO DE <i>GANTT</i> PARA A SOLUÇÃO.	85
FIGURA 30 – REFERENCIAS DE PROCESSOS PARA A SOLUÇÃO.....	85
FIGURA 31 – IDENTIDADE VISUAL.	86
FIGURA 32 – TELA DE CADASTROS.	87
FIGURA 33 – TELA DE CONFIGURAÇÃO DA INTEGRAÇÃO.	88
FIGURA 34 – TELA DE PARAMETRIZAÇÃO DO ALGORITMO.	88
FIGURA 35 – TELA DE VISÃO DO PLANEJAMENTO INTELIGENTE.....	89
FIGURA 36 – TELA DE DASHBOARD DO PLANEJAMENTO.	89
FIGURA 37 – PROTÓTIPO FINAL DO SISTEMA <i>OPTIMIZIA</i>	90
FIGURA 38 – PROTÓTIPO FINAL DO SISTEMA <i>OPTIMIZIA</i> COM IDENTIDADE VISUAL..	90
FIGURA 39 – SERVIÇO DE INTEGRAÇÃO COM ERP.	91
FIGURA 40 – ARQUITETURA DO SERVIDOR E BANDO DE DADOS.....	92
FIGURA 41 – ESCRITA DO ALGORITMO EM BACK-END.....	93
FIGURA 42 - FORMATO DA BASE DE DADOS FORNECIDA.....	93
FIGURA 43 – BASE DE DADOS <i>MS_TMP</i> NO FORMATO CORRETO UTILIZANDO A BASE FORNECIDA PELA PRODUZA.....	95
FIGURA 44 – VALIDAÇÃO DAS ORDENS DE PRODUÇÃO QUE POSSUEM MAIS DE UMA MÁQUINA PARA A BASE DE DADOS <i>MS_TMP</i>	95
FIGURA 45 – BASE DE DADOS <i>PT_TMP</i> NO FORMATO CORRETO UTILIZANDO A BASE FORNECIDA PELA PRODUZA.	96
FIGURA 46 – VALIDAÇÃO DAS ORDENS DE PRODUÇÃO QUE POSSUEM MAIS DE UMA MÁQUINA PARA A BASE DE DADOS <i>PT_TMP</i>	96
FIGURA 47 – BASE DE DADOS <i>JOB_PRIORITY_DUEDATE_TMP</i> ORIUNDA DA PLANILHA DA EMPRESA PRODUZA.....	97
FIGURA 48 – COMPARATIVO PARA VALIDAR PARTE DA BASE DE DADOS <i>JOB_PRIORITY_DUEDATE_TMP</i> . OBSERVA-SE COMO OS VALORES PARA DATA DA BASE E DIFERENÇA NA PLANILHA SÃO IGUAIS.....	98

FIGURA 49 – GRÁFICO DE <i>GANTT</i> PARA A PRIMEIRA SOLUÇÃO <i>BEST_LIST[0]</i> DA FRONTEIRA DE SOLUÇÕES GERADA.	100
FIGURA 50 – VISÃO AMPLIADA DO CALENDÁRIO DE <i>GANTT</i> , PARA FACILITAR A IDENTIFICAÇÃO DAS ORDENS.....	101
FIGURA 51 – VALIDAÇÃO DA FREQUÊNCIA DE AGENDAMENTO DE CADA ORDEM NA RESPOSTA.....	101
FIGURA 52 – ARQUITETURA DO <i>FRONT-END</i>	103
FIGURA 53 – <i>FRONT-END</i> DO SISTEMA <i>OPTIMIZIA</i>	104
FIGURA 54 – <i>DASHBOARD</i> DO SISTEMA <i>OPTIMIZIA</i>	105

LISTA DE TABELAS

TABELA 1 – MÁQUINAS ALTERNATIVAS E TEMPOS POR PROCESSOS PARA OPERAÇÃO VJ.....	49
TABELA 2 – HORÁRIO DE INÍCIO E TÉRMINO DE CADA OPERAÇÃO (HORA DE INÍCIO, HORA DE TÉRMINO).....	50
TABELA 3 – CLASSIFICAÇÃO SISTEMAS APS.....	64
TABELA 4 – CLASSIFICAÇÃO DOS USUÁRIOS SOBRE SOFTWARES APS.....	64
TABELA 5 – AVALIAÇÃO DE CARACTERÍSTICAS DOS SOFTWARES DE APS.....	64
TABELA 6 – CARACTERÍSTICAS MAIS IMPORTANTES EM SOFTWARES APS.....	65
TABELA 7 – RECURSOS MAIS BEM AVALIADOS EM SOFTWARES APS.....	65
TABELA 8 – FUNCIONALIDADES MAIS IMPORTANTES DE SISTEMAS APS.	65
TABELA 9 - ROTEIROS POSSÍVEIS DE FABRICAÇÃO.	94

LISTA DE QUADROS

QUADRO 1 – ESTUDOS SOBRE O PROBLEMA DE APS.	30
QUADRO 2 - TIPOS DE DADOS, DESCRIÇÃO E EXEMPLOS.	42
QUADRO 3 – FUNCIONALIDADES MAIS E MENOS AVALIADAS PELOS USUÁRIOS.....	66
QUADRO 4 – RESTRIÇÕES PARA ANÁLISE DA CAPACIDADE FINITA.	68
QUADRO 5 – FUNÇÕES PARA ANÁLISE DO ALGORITMO DE INTELIGÊNCIA ARTIFICIAL.	69
QUADRO 6 – DADOS DO <i>ERP</i>	72
QUADRO 7 – DADOS DO <i>MES</i>	73

LISTA DE SIGLAS

AG – Algoritmo Genético

NSGA-II – *Nondominated Sorting Genetic Algorithm II* - Algoritmo Genético de Classificação Não Dominada II

AI – *Artificial Intelligence* – Inteligência Artificial

APS – *Advanced Planning Systems* ou *Advanced Planning and Scheduling* – Sistema de Planejamento e Agendamento Avançado da Produção

ATP – *Available to Promise* – Disponibilidade para Promessa

BOM – *Bill of Materials* – Lista de Materiais

CRP – *Capacity Requirements Planning*

CTP – *Capable to Promise* – Capacidade para Promessa

DSR – *Design Science Research*

ERP – *Enterprise Resource Planning* – Sistema de Planejamento de Recursos Empresariais

FCS – *Finite Capacity Scheduling* – Sistema de Agendamento de Capacidade Finita

IoT – *Internet of Things* – Internet das Coisas

MES – *Manufacturing Execution System* – Sistema de Execução da Manufatura

ML – *Machine Learning* – Aprendizado de Máquina

MPS – *Master Plan and Scheduling* – Plano Mestre da Produção

MRP – *Material Requirements Planning* – Planejamento de Requisição de Materiais

MRP II – *Manufacturing Resource Planning* – Planejamento de Recursos de Manufatura

MVP – *Minimum Viable Product* – Produto Mínimo Viável

WIP – *Work-in-process* – Materiais em Processo

CONTEÚDO

1. INTRODUÇÃO	18
1.1. CONTEXTUALIZAÇÃO.....	19
1.2. FORMULAÇÃO DO PROBLEMA.....	22
1.3. OBJETIVO GERAL	23
1.4. OBJETIVOS ESPECÍFICOS	23
1.5. PRODUTO TECNOLÓGICO.....	23
1.6. CONTRIBUIÇÕES DO TRABALHO.....	24
1.6.1. Contribuições para a academia	25
1.6.2. Contribuições para a indústria	25
2. FUNDAMENTAÇÃO TEÓRICA	27
2.1. <i>ADVANCED PLANNING AND SCHEDULING (APS)</i>	27
2.2. PROBLEMAS DE PLANEJAMENTO E PROGRAMAÇÃO DE PRODUÇÃO	29
2.3. ARQUITETURA DE UM SISTEMA APS	33
2.3.1. Projeto de rede	36
2.3.2. Planejamento de demanda	37
2.3.3. Planejamento de rede de suprimentos	37
2.3.4. Cumprimento de demanda e <i>ATP</i>	38
2.3.5. Planejamento de distribuição e transporte	39
2.3.6. Gestão de inventário.....	39
2.3.7. Programação detalhada de produção.....	40
2.4. ALGORITMOS INTELIGENTES	41
2.4.1. <i>Machine Learning</i>	42
2.4.1.1. <i>Supervisionada</i>	43
2.4.1.2. <i>Não supervisionada</i>	44
2.4.1.3. <i>Semi supervisionada</i>	44
2.4.1.4. <i>Reforço</i>	44
2.4.2. Algoritmos evolutivos.....	45
2.4.2.1. <i>Algoritmo genético</i>	46
2.4.2.2. <i>Algoritmos genéticos de múltiplos objetivos</i>	54
2.5. PROPOSIÇÕES	56
3. ABORDAGEM METODOLÓGICA	58
4. DESENVOLVIMENTO.....	61
4.1. ETAPA 1 – COMPREENSÃO DO PROBLEMA	61

4.2. ETAPA 2 – GERAÇÃO DE ALTERNATIVAS	62
4.2.1. <i>Brainstorming</i>	62
4.2.2. <i>DSR</i> parcial sobre <i>APS</i>	63
4.2.3. Estudo de Mercado.....	63
4.2.4. <i>Mockup</i> do Sistema	66
4.3. ETAPA 3 – DESENVOLVIMENTO DO ARTEFATO.....	67
4.3.1. Definição da função objetivo.....	68
4.3.2. Coleta de Dados e Arquitetura do sistema	70
4.3.2.1. <i>Serviço de Integração de Dados</i>	72
4.3.2.2. <i>Servidor do sistema</i>	74
4.3.2.3. <i>Serviço de front-end</i>	74
4.3.3. Estruturação do algoritmo inteligente	75
4.3.4. Desenvolvimento da Solução (<i>design, front-end e back-end</i>).....	80
4.3.4.1. <i>Desenvolvimento da Solução – Design</i>	80
4.3.4.2. <i>Desenvolvimento da Solução - Back-end</i>	91
4.3.4.2.1. <i>Back-end – Tecnologias e Arquitetura</i>	91
4.3.4.2.2. <i>Detalhamento do Algoritmo do Sistema</i>	93
4.3.4.3. <i>Desenvolvimento da Solução - Front-end</i>	102
4.4. ETAPA 4 – AVALIAÇÃO	105
4.5. ETAPA 5 – CONCLUSÕES	106
5. RESULTADOS E DISCUSSÃO	108
5.1. REVISÃO BIBLIOGRÁFICA SISTEMÁTICA (RBS).....	108
5.2. <i>DESIGN SCIENCE RESEARCH (DSR)</i>	111
5.3. PRODUTO TECNOLÓGICO.....	112
6. CONSIDERAÇÕES FINAIS	114
6.1. DELIMITAÇÕES DO TRABALHOS.....	114
6.2. SUGESTÕES DE TRABALHOS FUTUROS	115
REFERÊNCIAS BIBLIOGRÁFICAS.....	117

1. INTRODUÇÃO

Os Sistemas Avançados de Planejamento e Programação da Produção (APS) são ferramentas de suporte à decisão que incluem diversos recursos, como a programação de capacidade finita e o planejamento baseado em restrições. Eles visam fornecer planos de produção viáveis para empresas, levando em conta fatores como a capacidade de recursos em diferentes níveis de decisão (estratégico, tático e operacional) e problemas de cadeia de suprimentos (DAVID, PIERREVAL, e CAUX, 2006).

Os sistemas APS utilizam algoritmos matemáticos avançados ou lógica para otimização ou simulação no agendamento (COX e BLACKSTONE, 2002). No entanto, segundo Vieira, Deschamps e Valle (2021), ainda não foram encontrados estudos que enfoquem o problema de APS com a aplicação de algoritmos de inteligência artificial.

A Inteligência Artificial (IA) é uma área da ciência e engenharia que se concentra na teoria e na prática do desenvolvimento de sistemas que apresentam características associadas à inteligência humana, como a resolução de problemas, planejamento, aprendizado e adaptação, percepção, compreensão de linguagens e ação no ambiente (TECUCI, 2012).

Desta forma, este trabalho aborda todas as etapas necessárias para a construção de um framework de um sistema avançado de planejamento de produção para o ambiente de manufatura, utilizando sistemas computacionais de inteligência artificial e conceitos da Indústria 4.0. Para facilitar a compreensão, foi dividido em cinco seções distintas.

Na seção 1, será apresentado uma contextualização sobre o tema, incluindo o problema, objetivo geral e específicos, produto tecnológico e contribuições para a academia e indústria. Na seção 2, será apresentada a fundamentação teórica sobre Sistemas Avançados de Planejamento e Programação da Produção, incluindo problemas endereçados, arquitetura de um sistema padrão, algoritmos matemáticos e inteligentes, de *machine learning* e de classificação, incluindo o algoritmo genético.

A metodologia utilizada é a *Design Science Research* e suas 5 etapas de desenvolvimento do produto tecnológico estarão descritas na seção 3. Na seção 4, serão apresentados os resultados alcançados com a realização deste trabalho, como a revisão bibliográfica sistemática, a aplicação da metodologia *design science research* e o produto tecnológico desenvolvido.

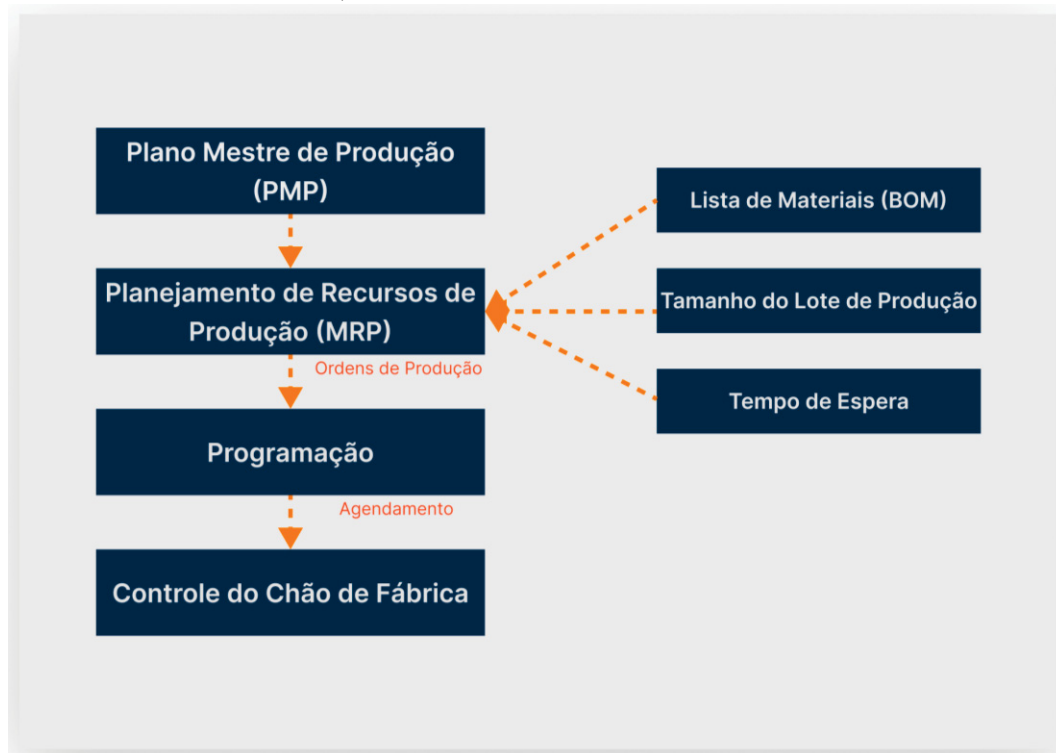
Finalmente, na seção 5, serão apresentadas as conclusões sobre todo o trabalho, incluindo novas perspectivas sobre os sistemas de APS com a aplicação de algoritmos inteligentes.

1.1. CONTEXTUALIZAÇÃO

Desde a década de 1970, a evolução dos sistemas de suporte à decisão na indústria tem sido constante, graças à disponibilidade crescente de computadores. Durante esse período, os sistemas de planejamento e controle de produção foram gradualmente aperfeiçoados, passando de sistemas de planejamento de requisitos de materiais (*Material Requirements Planning - MRP*), para sistemas de planejamento de recursos de manufatura (*Manufacturing Resource Planning - MRP II*), planejamento de recursos empresariais (*Enterprise Resource Planning - ERP*) e sistemas com capacidade finita (*Finite Capacity Scheduling - FCS*), os sistemas integrados de controle da produção (*Manufacturing Execution Systems – MES*) e culminando em sistemas avançados de planejamento e programação da produção (*Advanced Planning Systems - APS*). Cada uma dessas evoluções representou uma melhoria no processo de planejamento e controle, com a incorporação de novas tecnologias e uma compreensão mais profunda dos processos de manufatura (HVOLBY e STEGER-JENSEN, 2010; ZATTAR, 2004).

Com exceção dos primeiros sistemas de ponto de reabastecimento da década de 1960, o Planejamento de Requisitos de Materiais (*MRP*) foi a primeira geração de sistemas sistemáticos de planejamento de materiais. O *MRP* foi construído em torno de um processador de lista de materiais (*Bill of Materials*) e a vantagem era a capacidade de explodir os componentes necessários para construir os itens acabados e resumir a necessidade de componentes individuais no volume total de pedidos. O *MRP* segue uma abordagem hierárquica de cima para baixo (FIGURA 1) e começa com o Plano Mestre de Produção (*Master Plan and Scheduling – MPS*), que geralmente considera a quantidade e a data dos pedidos de produtos acabados. O *MPS* é então traduzido em datas de início e término, planejadas especificamente para todos os subconjuntos e componentes com base na estrutura do produto, e, em seguida, um problema detalhado de programação é resolvido para cumprir essas datas de vencimento. (HVOLBY e STEGER-JENSEN, 2010; CHEN e JI, 2007).

FIGURA 1 – O ESQUEMA DE MRP TRADICIONAL E AGENDAMENTO.



FONTE: Adaptado de Chen e Ji (2007).

Posteriormente, o *MRP* foi aprimorado para lidar com o planejamento de requisitos de capacidade e foi chamado de "*MRP* de Malha Fechada". Ele ofereceu *feedback* sobre informações que permitiram ajustar planos e a regeneração, e passou a ser conhecido como planejamento de recursos da manufatura ou *MRP II* (HVOLBY e STEGER-JENSEN, 2010). Em 1990, o *Gartner Group* (WYLIE, 1990) cunhou o termo Planejamento de Recursos Empresariais (*ERP*), pois os *softwares* foram integrados a outras áreas, como previsão, planejamento de longo prazo e gerenciamento de recursos críticos (HVOLBY e STEGER-JENSEN, 2010).

O *ERP* pode fornecer planos iniciais, mas frequentemente não tem ferramentas que ajudem o planejador a analisar e atualizar o planejamento. Para preencher esta limitação na funcionalidade do *ERP*, empresas de *software* lançaram sistemas e módulos avançados de planejamento e controle de produção no mercado desde a década de 1990 (DE MAN e STRANDHAGEN, 2018).

Estas ferramentas surgiram devido a três restrições no sistema *ERP*: 1) O *MRP* apenas faz a programação regressiva sem restrições de capacidade; 2) o *ERP* não transforma o *MRP* em um cronograma de produção detalhado; 3) o *ERP* limita a

visualização da capacidade agregada necessária e das programações diárias de produção (DE MAN e STRANDHAGEN, 2018).

Contrariamente aos sistemas *ERP* convencionais, os sistemas de Programação com Capacidade Finita (*FCS*) levam em conta a capacidade produtiva como restrição principal para a tomada de decisões, garantindo que a programação de produção seja viável (ZATTAR, 2004). Já os sistemas avançados de planejamento e programação da produção (*APS*) buscam encontrar planos viáveis próximos da otimização, considerando gargalos potenciais explicitamente (STADTLER e KILGER, 2005). Muitos sistemas *ERP* e *APS* permitem incluir fornecedores e clientes no processo de planejamento, otimizando a cadeia de suprimentos em tempo real (WORTMANN, 1998; KENNERLY e NEELY, 2001). Por outro lado, o Sistema de Execução da Manufatura (*MES*) evoluiu para auxiliar na execução da produção, monitoramento e controle no chão de fábrica, superando as desvantagens dos sistemas *ERP* na exploração de informações em tempo real do ambiente produtivo. O *MES* controla as operações que permitem a realização dos planos do sistema *ERP* (ARICA, 2014).

O *APS* supera muitas das funcionalidades de planejamento e programação do sistema *ERP*, tornando-se uma ferramenta importante e eficiente de planejamento e controle. Uma de suas vantagens é a capacidade de simular diferentes cenários de planejamento antes de definir o plano final. O *APS* complementa, mas não substitui, os sistemas *ERP* existentes. Enquanto o sistema *ERP* trata das atividades básicas, como pedidos de clientes e contabilidade, o sistema *APS* se concentra nas atividades diárias de análise e suporte à tomada de decisão (HVOLBY e STEGER-JENSEN, 2010).

Na literatura existem diversas formas de se referir aos sistemas de *APS* (*Advanced Planning and Scheduling*), como por exemplo: planejamento e programação avançada, planejamento avançado, programação avançada, sistema de planejamento e programação da produção, sistema avançado de planejamento e programação da produção, sistema de planejamento e controle da produção avançado, sistema de planejamento da produção avançado, sistema de planejamento de produção e materiais avançado, e sistema de programação da produção avançado. Contudo, neste trabalho, será utilizado o termo Sistemas Avançados de Planejamento e Programação da Produção ou simplesmente *APS* para referir-se a esses sistemas.

1.2. FORMULAÇÃO DO PROBLEMA

De acordo com a pesquisa de Vieira, Deschamps e Valle (2021), os sistemas de planejamento da produção estão atrasados em relação aos últimos avanços na tecnologia, especialmente no que diz respeito à digitalização, transformação digital e Indústria 4.0, com a utilização de *big data*, nuvem, automação, *IoT* e algoritmos avançados. Embora a pesquisa tenha encontrado algumas abordagens que incluem *IoT* e *RFID*, essas aplicações ainda são limitadas.

Os autores destacam que a participação humana ainda é essencial para o correto funcionamento de um sistema de planejamento da produção (APS). Esses sistemas dependem de dados precisos, que precisam ser gerados com base em cenários precisos. O gestor deve ter conhecimento aprofundado da fábrica e do processo para aproveitar plenamente as capacidades do APS. Muitas vezes, esses gestores usam planilhas para apoiar o APS, uma vez que os sistemas ainda são complexos e pouco flexíveis. Essas planilhas permitem uma maior flexibilidade para inserir e modificar dados, dando ao gestor mais autonomia.

Além disso, existem muitos métodos, abordagens, algoritmos e modelos matemáticos que têm sido considerados pelos pesquisadores nos últimos anos, mas ainda não foram encontrados estudos que enfoquem o problema do APS com a utilização de algoritmos de inteligência artificial.

Dessa forma, Vieira, Deschamps e Valle (2021) concluem que ainda há uma grande oportunidade para pesquisas futuras, especialmente no contexto da Indústria 4.0 e o APS. Por exemplo, como desenvolver sistemas mais flexíveis e intuitivos para os gestores? Como aplicar o *Big Data* ao APS para obter informações úteis para o gestor? Como projetar uma arquitetura de nuvem que atenda às necessidades de cada empresa sem afetar outros sistemas? Como usar a inteligência artificial para tornar o APS mais autônomo e reduzir a dependência das pessoas? Que aplicativos *IoT* e *RFID* podem ser integrados ao APS para troca de informações em tempo real? Como criar um sistema que possa gerar cenários em tempo real, considerando interferências no processo de trabalho?

Com base nesta pesquisa e problemas observados, como é possível desenvolver um *framework* de um sistema de sistemas avançados de planejamento e programação da produção que utilize algoritmos inteligentes, gerando flexibilidade e autonomia em sua operação e que seja replicado em larga escala para as empresas?

1.3. OBJETIVO GERAL

O objetivo é desenvolver um *framework* de um sistema avançado de planejamento de produção para o ambiente de manufatura, utilizando sistemas computacionais de inteligência artificial e aplicando conceitos da Indústria 4.0.

1.4. OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo principal, foram estabelecidas as seguintes metas específicas:

- 1) Identificar na literatura como os sistemas avançados de planejamento e programação da produção (*APS*) são abordados;
- 2) Selecionar o algoritmo de Inteligência Artificial mais apropriado para solucionar problemas de *APS*;
- 3) Projetar o *framework* do sistema de *APS*;
- 4) Avaliar a aplicabilidade do algoritmo em um problema real de *APS*.

1.5. PRODUTO TECNOLÓGICO

O produto tecnológico apresentado neste trabalho é um *MVP* de um produto comercial de *APS* denominado de *OptimizIA* que foi concebido para solucionar os problemas de agendamento e programação de ordens de produção através do uso de algoritmos de inteligência artificial. Para a concepção deste *framework*, a metodologia de *Design Science Research (DSR)* será utilizada. Conforme FIGURA 2, o produto tecnológico será dividido em quatro etapas principais:

- 1) Definição das funções objetivas do produto e dos algoritmos inteligentes, além dos dados necessários para a realização dos cálculos e simulações.
- 2) Coleta, tratamento e estruturação dos dados a partir dos sistemas de planejamento de recursos empresariais (*ERP*) e de execução da manufatura (*MES*), incluindo a discussão sobre a integração destes sistemas e o desenvolvimento da arquitetura do sistema.

- 3) Apresentação e discussão de um algoritmo de inteligência artificial para a geração do melhor cenário de programação em tempo real e resolução de problemas de *Advanced Planning and Scheduling (APS)*.
- 4) Disponibilização de informações para equipes gerenciais e de chão de fábrica.

A metodologia *DSR* é focada na criação de soluções tecnológicas para problemas práticos e na avaliação de sua eficácia na prática. Neste trabalho, ela será utilizada para desenvolver um produto tecnológico que otimize a produção, por meio do uso de algoritmos inteligentes e dos dados coletados dos sistemas *ERP* e *MES*, além de apresentar informações para equipes gerenciais e de chão de fábrica. O objetivo final é otimizar a produção e melhorar a eficiência.

FIGURA 2 – PRODUTO TECNOLÓGICO.



FONTE: O Autor (2023).

1.6. CONTRIBUIÇÕES DO TRABALHO

A pesquisa e desenvolvimento de um novo sistema comercial de Programação Avançada da Produção (*APS*) utilizando inteligência artificial apresenta diversas contribuições tanto para a academia quanto para a indústria. Na academia, a pesquisa pode trazer novas abordagens e técnicas para a área de *APS* e inteligência artificial, além de contribuir para a formação de novos pesquisadores e profissionais capacitados para atuarem nesse campo. Já na indústria, o desenvolvimento de um sistema *APS* com algoritmos inteligentes pode trazer inúmeras melhorias, desde a otimização da

produção e redução de custos, até o aumento da qualidade e da eficiência dos processos produtivos. Além disso, o sistema também pode auxiliar na tomada de decisões estratégicas e no planejamento de longo prazo das empresas, tornando-as mais competitivas no mercado.

1.6.1. Contribuições para a academia

Um novo sistema comercial de Programação Avançada da Produção (*APS*) utilizando Inteligência Artificial (*IA*) pode trazer diversas contribuições para a academia. Em primeiro lugar, pode ser uma oportunidade para desenvolver pesquisas mais avançadas em *IA* aplicada à otimização de processos produtivos. Além disso, pode gerar novos estudos sobre o uso de algoritmos de otimização multiobjetivo, como o *NSGA-II*, em sistemas de *APS*.

Outra contribuição importante pode ser o estudo das possibilidades de utilização de outras técnicas de *IA* em sistemas de *APS*, como Redes Neurais Artificiais (*RNAs*) e Aprendizado de Máquina (*Machine Learning*), para melhorar a eficiência e efetividade do sistema. O desenvolvimento de um novo sistema comercial de *APS* também pode ser uma oportunidade para o estudo de melhores práticas em desenvolvimento de *software* para sistemas de alta complexidade e escalabilidade.

Por fim, o desenvolvimento de um novo sistema comercial de *APS* utilizando *IA* pode ser uma contribuição relevante para o mercado de produção industrial. Com a crescente demanda por sistemas cada vez mais eficientes e automatizados, a criação de um sistema que atenda às necessidades do mercado pode ser uma oportunidade para estudos de casos e análises de impacto econômico e social.

1.6.2. Contribuições para a indústria

Um novo sistema comercial de programação avançada da produção utilizando inteligência artificial pode trazer diversas contribuições para a indústria. Através da utilização de algoritmos inteligentes, é possível melhorar a eficiência e efetividade da produção, reduzir custos operacionais, aumentar a qualidade dos produtos e serviços oferecidos e, conseqüentemente, aumentar a competitividade no mercado. Além disso, o sistema pode ajudar as empresas a lidar com os desafios de produção em larga

escala, como a necessidade de balancear recursos, cumprir prazos e lidar com imprevistos, permitindo uma tomada de decisão mais rápida e precisa.

Além das contribuições mencionadas anteriormente, o sistema de programação avançada da produção utilizando inteligência artificial também pode trazer outros benefícios para a indústria. Por exemplo, ao otimizar o tempo de produção e minimizar o tempo ocioso das máquinas, o sistema pode aumentar a produtividade da empresa e reduzir seus custos operacionais. Isso pode levar a uma maior competitividade da empresa no mercado e um aumento na sua rentabilidade.

O sistema pode proporcionar uma maior flexibilidade e adaptabilidade à empresa. Com a capacidade de otimizar a programação da produção em tempo real, a empresa pode responder rapidamente a mudanças na demanda do mercado ou na disponibilidade de recursos. Isso pode permitir que a empresa se adapte mais facilmente a diferentes situações e se mantenha competitiva em um ambiente de negócios em constante mudança.

Por fim, um sistema de programação avançada da produção também pode trazer contribuições na área de sustentabilidade, permitindo uma produção mais eficiente e responsável do ponto de vista ambiental.

2. FUNDAMENTAÇÃO TEÓRICA

A presente fundamentação teórica do trabalho foi baseada na comparação de diferentes concepções de *APS* por diversos pesquisadores. Em seguida, foram abordados os problemas de planejamento e programação da produção que justificam a existência de um sistema *APS*. Depois, houve uma ênfase na arquitetura de um sistema *APS* convencional, incluindo suas funcionalidades, entradas de dados e saídas. Por fim, a fundamentação teórica foi concentrada no estudo do algoritmo de inteligência artificial, como aprendizado de máquina e algoritmos genético.

2.1. *ADVANCED PLANNING AND SCHEDULING (APS)*

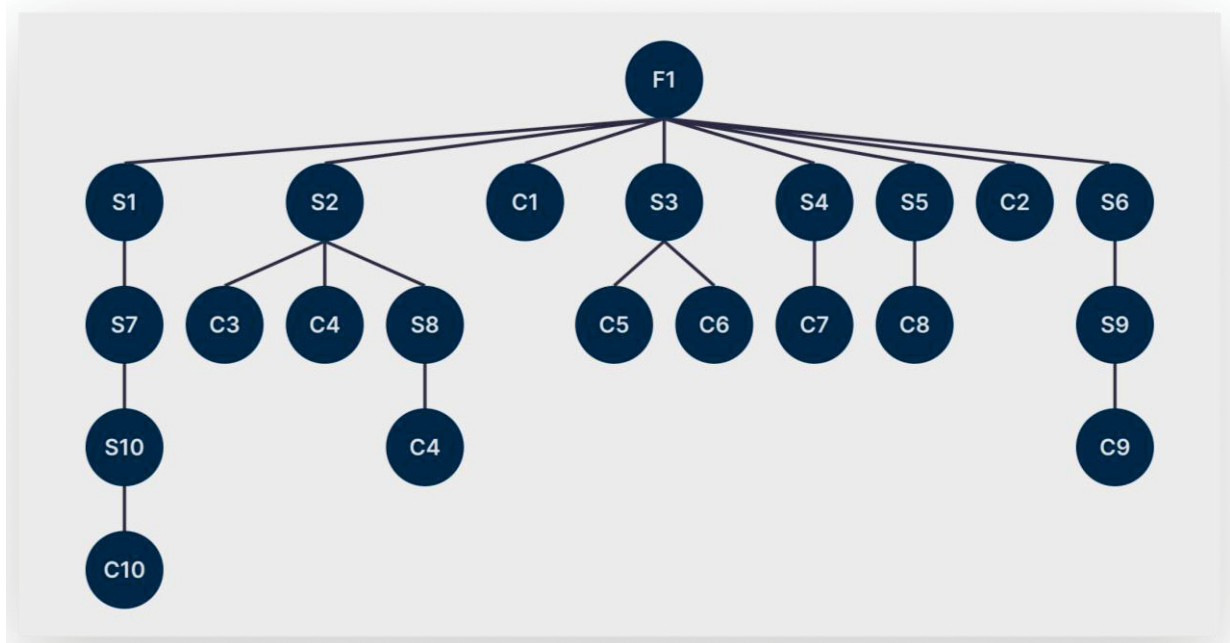
Muitas empresas de manufatura possuem produtos com uma estrutura multinível, ou seja, um produto final compreende vários subconjuntos e componentes, e cada subconjunto também pode exigir subconjuntos e componentes (FIGURA 3). O planejamento e programação avançados (*APS*) trata da alocação eficaz dos recursos de produção para completar esses produtos, de modo que as restrições de produção sejam satisfeitas e os objetivos de produção sejam atendidos (MOON, SEO, YUN e GEN, 2006; LIN, HWANG e WANG, 2007; CHEN, JI e WANG, 2011). Além disso, o *APS* também faz planos e cronogramas, levando em consideração a disponibilidade de materiais, mão de obra e máquinas por meio de processos de planejamento e programação hierárquicos para lidar com compensações complexas entre prioridades concorrentes (MOON, KIM e GEN, 2004; ZHONG *et al.*, 2013).

Advanced Planning Systems (APS), também conhecido como *Advanced Planning and Scheduling*, é definido no *APICS Dictionary* (COX e BLACKSTONE, 2002) como "qualquer programa de computador que usa algoritmos matemáticos avançados ou lógica para realizar otimização ou simulação em agendamento de capacidade finita. Essas técnicas consideram simultaneamente uma variedade de restrições e regras de negócios para fornecer recursos de planejamento e programação em tempo real, suporte à decisão, disponibilidade para promessa (*ATP*) e capacidade para promessa (*CTP*)" (LIN, HWANG e WANG, 2006).

Os sistemas avançados de planejamento e programação da produção referem-se a um processo de gerenciamento de produção em que as matérias-primas e a

capacidade de produção são alocadas de forma otimizada para atender a diferentes objetivos, como a maximização da produção da planta (LEE, JEONG, & MOON, 2002).

FIGURA 3 – ESTRUTURA DE PRODUTO COM CINCO NÍVEIS.



FONTE: Adaptado de Chen e Ji (2007).

Apesar da funcionalidade da cadeia de suprimentos, a maioria das implementações de *Advanced Planning and Scheduling (APS)* são limitadas a uma única organização ou a um único local de fabricação. No entanto, o APS leva em consideração as restrições tanto no nível da empresa quanto no nível da fábrica. As questões de materiais e capacidade são consideradas simultaneamente e as questões de fabricação, distribuição e transporte são integradas (HVOLBY e STEGER-JENSEN, 2010). Assim, o APS se adapta ao ambiente complexo da manufatura, onde abordagens simples de planejamento e programação não são capazes de lidar com as interações entre diferentes entidades concorrentes (ZHONG *et al.*, 2013; CHEN e JI, 2007).

Em diferentes empresas, o APS é altamente personalizado devido às diversas situações e requisitos, que são os principais obstáculos para a aplicação de um APS genérico (HVOLBY e STEGER-JENSEN, 2010; ZHONG *et al.*, 2013). No campo da pesquisa, os estudos de APS geralmente se concentram nas perspectivas teóricas, incluindo a complexidade do modelo, o escopo do problema e o design de algoritmos de solução, enquanto os relatos sobre casos reais são muito limitados (CHEN e JI, 2007; MEYR *et al.*, 2008). Além disso, outros estudos indicam que existem vários

problemas envolvidos no uso de software de planejamento, como a alta complexidade, a falta de treinamento e conhecimento entre gerentes e funcionários, a baixa precisão de dados e a falta de suporte do fornecedor do *software* (PETRONI, 2002).

2.2. PROBLEMAS DE PLANEJAMENTO E PROGRAMAÇÃO DE PRODUÇÃO

O objetivo do sistema *APS* é solucionar o problema de planejamento da produção, e durante décadas, a academia frequentemente abordou esses problemas como sendo determinísticos e estáticos, formulando diversos modelos matemáticos para sua resolução (LIN, HWANG e WANG, 2006).

Segundo Chen, Ji e Wang (2011), o problema *APS* leva em consideração explicitamente as datas de vencimento dos produtos, as sequências de operações entre os itens e as restrições de capacidade do sistema de fabricação. O objetivo é encontrar o menor custo de tempo ocioso de produção e multas por atraso ou antecipação de entrega de um pedido.

Por outro lado, Lee, Jeong e Moon (2002) argumentam que em um ambiente de cadeia de suprimentos, o problema de produção é que uma série de pedidos específicos do cliente precisa ser fabricada em um ambiente de múltiplos projetos. Esses pedidos específicos do cliente estão sujeitos a prazos apertados, enquanto ao mesmo tempo existem períodos longos de duração, que resultam de peças específicas do pedido que precisam ser fabricadas internamente ou adquiridas externamente (KOLISCH e HESS, 2000).

O problema de planejamento e programação avançados (*APS*) tem recebido grande atenção das pesquisas nos últimos anos. No entanto, poucos resultados estão disponíveis em casos do mundo real (CHEN, JI e WANG, 2011). No QUADRO 1, apresentamos uma lista de alguns pesquisadores que abordaram o problema de planejamento e programação avançada (*APS*) e objetivos que buscavam em seus estudos.

O problema de *APS* é considerado NP-difícil (BILLINGTON, MCCLAIN e THOMAS, 1983; MOON, SEO, YUN e GEN, 2006), o que significa que qualquer abordagem de otimização exata é altamente impossível de resolver este tipo de problema de forma eficiente. Por esse motivo, métodos heurísticos e algoritmos

genéticos (AGs) são frequentemente utilizados para encontrar boas soluções para problemas de APS (CHEN, JI e WANG, 2011).

QUADRO 1 – ESTUDOS SOBRE O PROBLEMA DE APS.

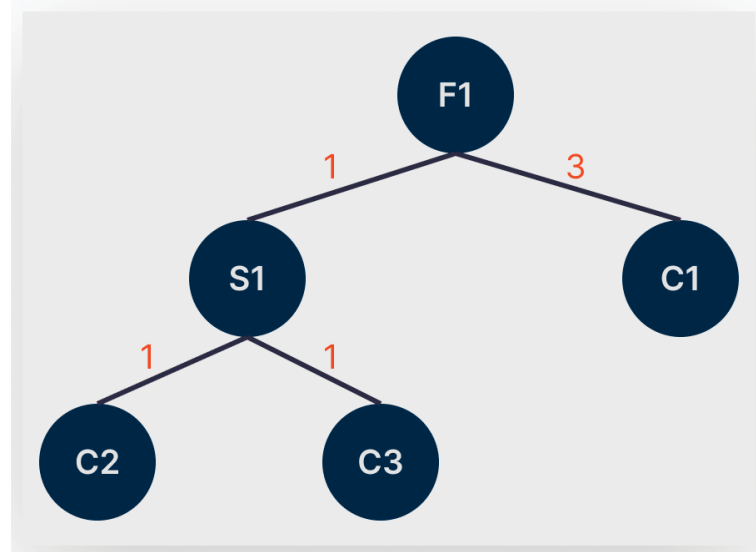
Foco	Pesquisador
Discute o problema de planejamento e programação do processo integrado para garantir que a produção seja eficiente e eficaz.	Palmer (1996); Sundaram e Fu (1988)
Considera o problema de programação de fluxo de fábrica com duas máquinas busca encontrar uma solução que minimize o tempo total de fluxo, considerando também o mínimo tempo para completar todas as tarefas (<i>makespan</i>).	Gupta, Neppalli e Werner (2001)
Discute que as vantagens de usar roteamentos alternativos de máquinas-ferramenta na oficina mecânica podem ser significativas. Essa abordagem pode reduzir o tempo de entrega e melhorar a utilização das máquinas. No entanto, quando há um grande número de trabalhos para agendar e muitas máquinas alternativas disponíveis, a utilização de técnicas matemáticas de programação para equilibrar a carga de trabalho pode ser proibitiva.	Hankins, Wysk e Fox (1984)
Sugere uma formulação alternativa do problema de programação de chão de fábrica, com base na ideia de que as restrições de capacidade e as relações de precedência de operação podem ser violadas, desde que se mantenha a data de vencimento como prioridade.	Holloway e Nelson (1974)
Analisa que o problema de atribuir uma data de vencimento comum a um conjunto de trabalhos e programá-los em uma única máquina pode ser complexo. Isso requer consideração dos tempos de processamento controláveis e uma situação em que todos os tempos de processamento possam ser reduzidos proporcionalmente. Além de atribuir uma data de vencimento comum, o objetivo é minimizar a soma das penalidades de adiantamento e atraso e também o número de trabalhos atrasados.	Biskup e Jahnke (2001)
O problema de minimizar o atraso máximo ou o número de trabalhos atrasados em um <i>flowshop</i> de duas máquinas também é importante. Isso requer consideração da data de vencimento e dos tempos de processamento.	Lin (2001)
Um problema de programação para um <i>flowshop</i> de duas máquinas que considera máquinas de processamento em lote, a medida de adiantamento/atraso e uma data de vencimento comum.	Sung e Min (2001)
Propõe uma forma de carregamento direto para agendar trabalhos na capacidade disponível.	Hastings, Marshall e Willis (1982)
Propõe-se uma nova estrutura para sistemas de planejamento e programação de manufatura, que formule um processo iterativo entre a rede de pedidos e a rede de operações para determinar o tamanho dos pedidos e a programação das operações.	Sum e Hill (1993)
Trata-se do problema geral de planejamento e programação, com o objetivo comum de minimizar o <i>makespan</i> , devido ao crescente interesse na estratégia de produção " <i>Just in Time</i> " (JIT) na indústria.	Cheng e Gupta (1989); Baker e Scudder (1990); Gordon, Proth e Chu (2002)
Explora-se um problema de produção de curto prazo para produtos com estrutura multinível, visando minimizar a soma ponderada de atrasos e precocidade dos itens. A pesquisa foi baseada no pressuposto da tecnologia de grupo (GT), que simplificou o problema integrado.	Kim e Kim (1996)
Propõe-se solução para o problema de máquinas e trabalhadores multifuncionais e/ou polivalentes, agrupados em estações de trabalho e	Liu, Wang, Leung e Li (2016)

células, por meio de problemas de configuração de programação de recurso duplo restrito (<i>DRC</i>).	
Revisa-se o <i>APS</i> a partir de aspectos teóricos de planejamento, programação e integração, destacando as abordagens baseadas em agentes. São identificadas as direções futuras, com abordagens inteligentes e em tempo real, utilizando tecnologias avançadas, como a fabricação sem fio.	Shen, Wang e Hao (2006)
Implementa-se o <i>APS</i> em uma indústria de conversão de alumínio, mas as decisões são baseadas em dados estáticos.	David, Pierreval e Caux (2006)
Propõe-se um <i>APS</i> para apoiar o planejamento da cadeia de suprimentos, considerando recursos limitados, elaborando decisões em três categorias (longo, médio e curto prazo), com base no projeto de rede dos módulos.	Neumann, Schwindt e Trautmann (2002)

FONTE: O autor (2023).

Para ilustrar o problema do APS, será utilizado um exemplo apresentado por Chen e Ji (2007). Considere um problema de planejamento de produção de produtos com estruturas de vários níveis. Uma simples estrutura de produto é apresentada na FIGURA 4, na qual o nó raiz representa o produto final (F1), composto por uma submontagem (S1) e um componente (C1).

FIGURA 4 – EXEMPLO SIMPLES DE UMA ESTRUTURA DE PRODUTO.

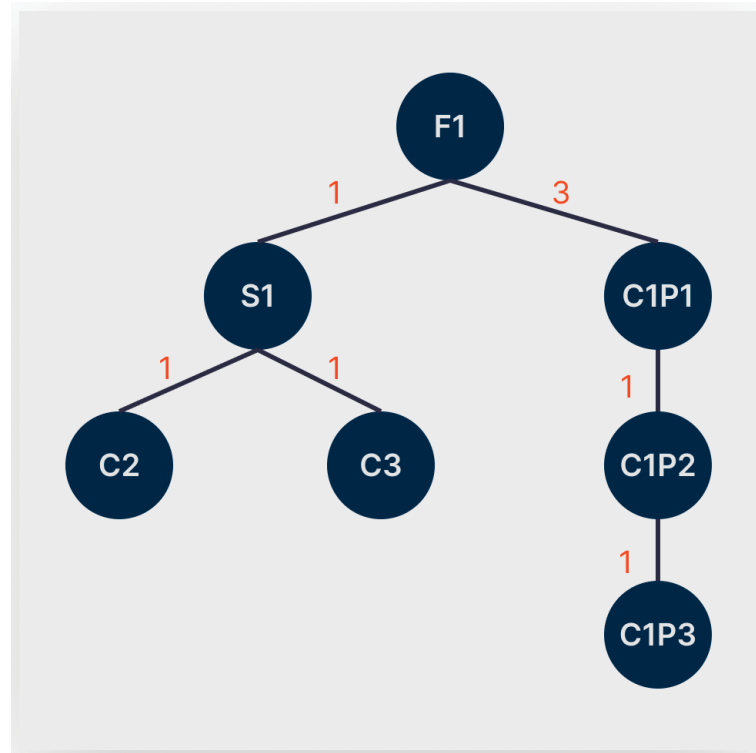


FONTE: Adaptado de Chen e Ji (2007).

A submontagem (S1) é formada pelos componentes C2 e C3. Esta estrutura de produto multinível é comum na indústria e geralmente é mais complexa. Nela, os itens (produtos finais, submontagens e componentes) possuem restrições de precedência, ou seja, os itens filhos precisam ser concluídos antes da produção dos itens pais. Cada item requer várias operações em máquinas qualificadas disponíveis para a produção. Portanto, é necessário considerar restrições de tempo de capacidade e fazer algumas suposições adicionais.

A estrutura do produto, os pedidos de itens e suas datas de vencimento são conhecidos com antecedência, assim como o tempo de processamento das operações. Uma estratégia lote por lote é adotada para a produção de itens e os tempos de preparação são insignificantes ou incluídos nos tempos de processamento. Se um item requer várias operações, ele é dividido em vários itens para refletir as operações. Por exemplo, se o item final C1 precisar passar por três operações (P1, P2, P3), ele será dividido em três itens filho: C1P1, C1P2, C1P3. Neste caso, C1P3 é um item filho de C1P2, C1P2 é um item filho de C1P1 e C1P1 é um item filho de F1, conforme mostrado na FIGURA 5. Para simplificar, só consideraremos produtos finais, subconjuntos e componentes. Além disso, cada operação pode ser processada por apenas uma máquina por vez, não é possível interrompê-la e cada máquina só pode trabalhar oito horas por dia.

FIGURA 5 – EXEMPLO DE UMA ESTRUTURA DE PRODUTO.

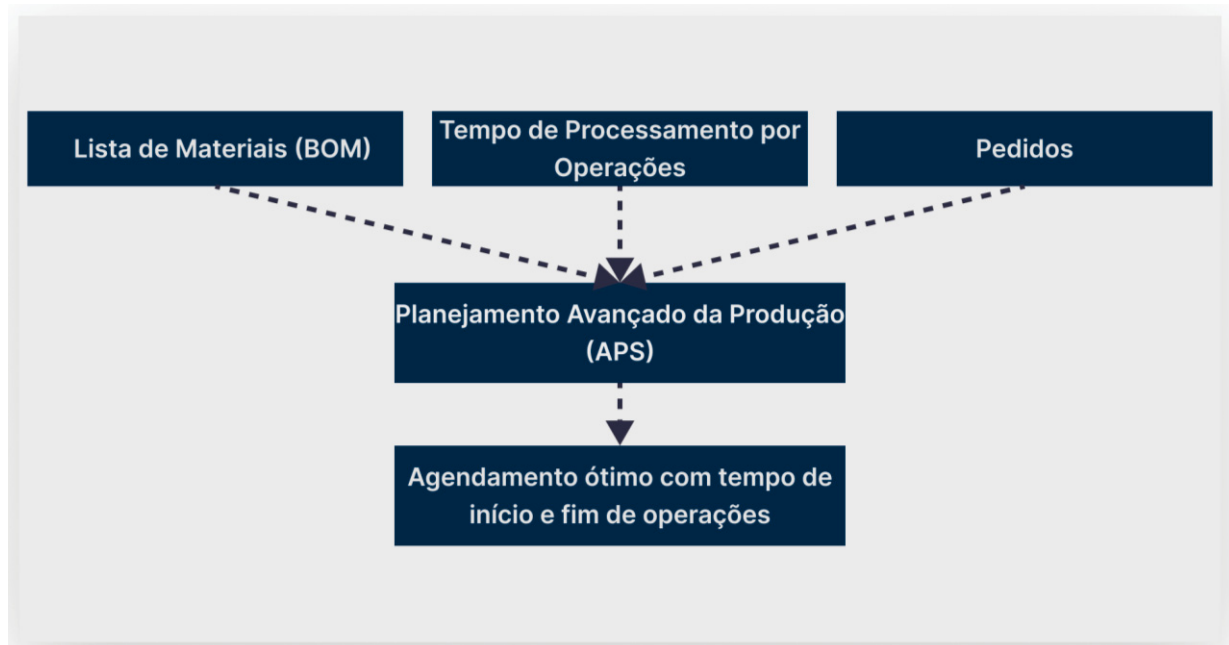


FONTE: Adaptado de Chen e Ji (2007).

O objetivo é encontrar um cronograma ideal para os pedidos de forma a minimizar o tempo ocioso da produção, as penalidades por atraso e antecipação. Minimizar o tempo ocioso da produção significa maximizar a utilização da máquina e reduzir o tempo de fluxo. Isso é importante porque reflete o *lead time* de fabricação e o nível de estoque *WIP* (*work-in-process*). Além disso, é desejável concluir todos os

trabalhos o mais próximo possível de suas datas de vencimento, já que a entrega antecipada ou tardia de um pedido resulta em custos elevados. Se um pedido for finalizado antes da data de vencimento, ele precisará ser mantido em estoque, resultando em penalidades de antecipação. Se for concluído após a data de vencimento, haverá multas por atraso, devido à insatisfação do cliente, contrato ou perda de reputação. O problema *APS* é caracterizado por atender às demandas dos clientes, reduzir o estoque de *WIP* e levar em consideração as restrições de capacidade de recursos e as complexas restrições de precedência entre as operações, como representado no diagrama esquemático na FIGURA 6.

FIGURA 6 – DIAGRAMA ESQUEMÁTICO DO PROBLEMA APS.



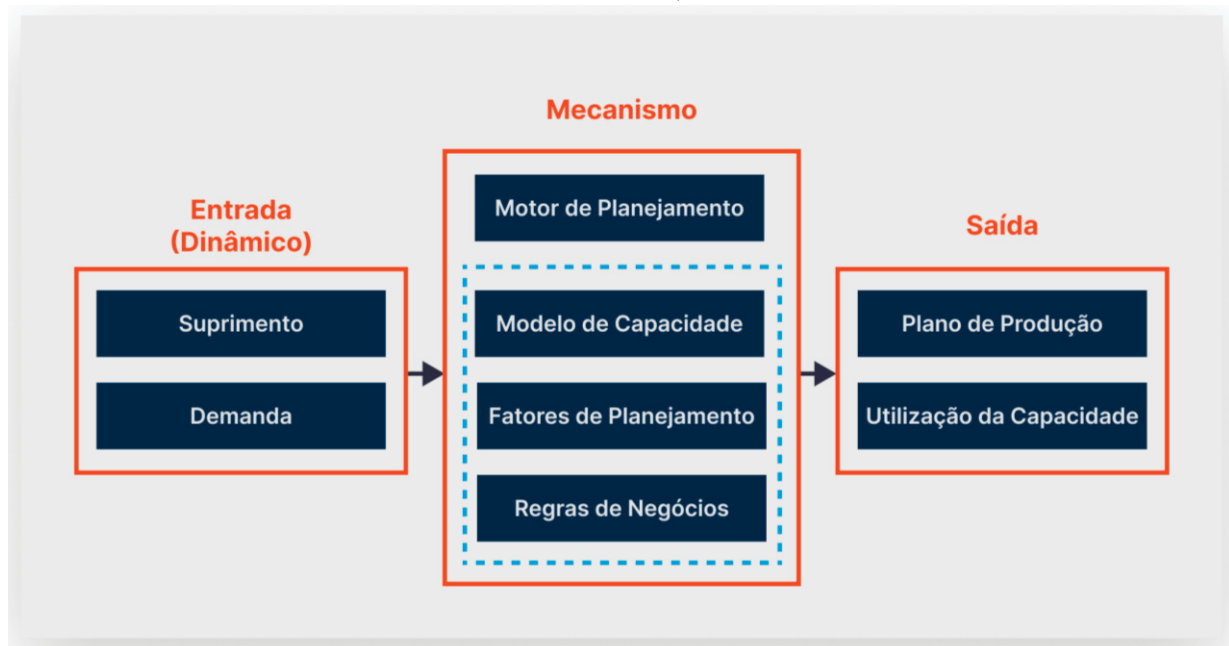
FONTE: Adaptado de Chen e Ji (2007).

2.3. ARQUITETURA DE UM SISTEMA APS

A arquitetura básica do *APS* (FIGURA 7) apresentada por Lin, Hwang e Wang (2006) é dividida em três partes. Na primeira fase, dois tipos de dados devem ser preparados no pré-processo: dados dinâmicos e estáticos. Os dados dinâmicos incluem informações sobre o suprimento (produtos acabados, material em processo) e demanda (pedidos confirmados, demanda apoiada pela alocação). Já os dados estáticos incluem o modelo de capacidade (estrutura do produto, lista de materiais, recursos-chave da máquina), fatores de planejamento (tempo de ciclo, eficiência) e regras de negócio.

Na segunda parte, é necessário construir um mecanismo para tomar decisões de planejamento com base no modelo de capacidade finita, conhecido como mecanismo de planejamento. E, por fim, na terceira parte, é necessário criar um pós-processo para organizar os resultados do planejamento do mecanismo.

FIGURA 7 – DIAGRAMA DE UMA ARQUITETURA BÁSICA DE APS.



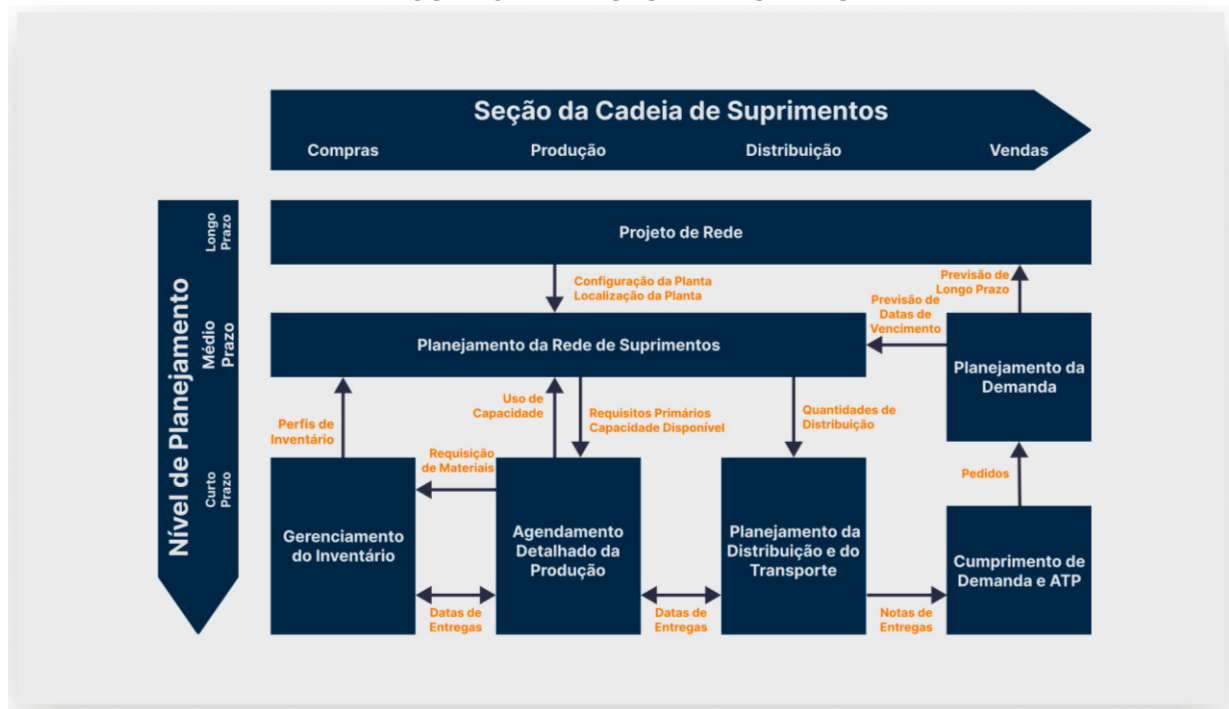
FONTE: Adaptado de Lin, Hwang e Wang (2006).

Os resultados do planejamento são apresentados em forma de plano de produção e uso da capacidade. O plano de produção mostra o tempo e a quantidade necessários para produzir um produto específico de acordo com a demanda associada. Já o uso da capacidade mostra o nível de utilização de cada grupo de máquinas-chave em qualquer intervalo de tempo dentro do horizonte de planejamento. Além disso, o APS possui uma função de simulação "e se", que permite avaliar diferentes cenários de demanda e capacidade.

Os objetivos alimentados no mecanismo de planejamento incluem minimizar a quantidade de pedidos não atendidos, atrasos nos pedidos em relação à data comprometida, data solicitada pelo cliente, a lacuna com a meta de produção, o estoque e o uso de recursos alternativos. As restrições são derivadas dos dados estáticos, como o modelo de capacidade, fatores de planejamento e regras de negócio. O lado da demanda e o lado da oferta serão equilibrados de maneira ideal. A solução APS, se bem formulada, deve fornecer um plano de produção preciso e confiável para a colaboração em toda a empresa (LIN, HWANG e WANG, 2006).

Uma característica essencial da solução APS é o modelo de capacidade finita. Todo o processo de Fabricação é modelado em termos de camada e cada camada é composta de algumas operações-chave com máquinas-chave anexadas. No modelo, cada produto deve seguir rota de processo predefinida para ocupar a capacidade finita da máquina. Além disso, a qualquer camada de um produto específico é atribuída um tempo de ciclo predefinido (LIN, HWANG e WANG, 2006).

FIGURA 8 – ESTRUTURA DE UM APS.



FONTE: Adaptado de Meyr, Wagner e Rohde (2000) e Stadtler (2005).

Da mesma forma, Neumann, Schwindt e Trautmann (2002) discutem a arquitetura do APS com base em vários autores. Para eles, em contraste com os sistemas de Planejamento de Recursos Empresariais (ERP), a limitação de recursos é levada em consideração em todas as fases do planejamento. Um planejamento monolítico da cadeia de suprimentos é inviável devido à complexidade e ao tamanho dos problemas de decisão individuais. Além disso, os diferentes horizontes de planejamento motivam uma divisão em planejamento de longo, médio e curto prazo, que consiste nas seções aquisição, produção, distribuição e vendas. Por isso, os sistemas avançados de planejamento e programação da produção são frequentemente decompostos em módulos, respeitando as interdependências entre eles.

A FIGURA 8 apresenta a estrutura de um APS genérico, composto por módulos de projeto de rede, planejamento de demanda, planejamento de rede de

abastecimento, programação de produção detalhada, planejamento de distribuição e transporte, atendimento de demanda e disponibilidade para promessa (*ATP*) e gerenciamento de estoque. Essa estrutura reflete a hierarquia de tarefas de planejamento comum, implementada por diferentes fornecedores de *software*. Cada caixa representa um módulo do *APS*, e os arcos entre as caixas ilustram os fluxos de informação entre os módulos (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.3.1. Projeto de rede

O projeto da rede de suprimentos se concentra na estrutura física da cadeia e, portanto, aborda questões relacionadas ao planejamento de localizações de fábricas, depósitos ou outras instalações de armazenamento, bem como o design de layouts de fábricas individuais. Normalmente, o horizonte de planejamento varia de três a dez anos (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

O objetivo é determinar o número ideal, localizações e capacidades de fábricas e armazéns para minimizar o custo total anualizado da rede. Este custo inclui despesas relacionadas ao estabelecimento de uma fábrica ou depósito em um local específico, custo de produção nas fábricas, custos de manuseio de material em depósitos e gastos com transporte. Diferentes tipos de restrições devem ser considerados, incluindo restrições de estrutura de rede, o que significa que um link entre uma fábrica e um depósito ou uma zona de cliente só pode existir se as instalações correspondentes forem construídas. Além disso, é necessário considerar restrições de fornecimento único (HOLMBERG, RÖNNQVIST e YUAN, 1999), que impõem que uma zona de cliente seja atendida por apenas um depósito. Restrições de equilíbrio de material garantem que as taxas de produção nas fábricas individuais correspondam ao fluxo de produtos para os armazéns e que a quantidade de um produto recebida por uma zona de cliente seja igual à demanda correspondente. Outras restrições incluem limites inferiores e superiores da capacidade de produção, armazenamento e transporte (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

O projeto da rede é baseado em mudanças a longo prazo nos programas de produtos, números de vendas e tecnologias de produção. Portanto, as principais entradas são previsões a longo prazo para vendas potenciais de grupos de produtos

em regiões específicas, resultantes do módulo de planejamento de demanda (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.3.2. Planejamento de demanda

O gerenciamento da cadeia de suprimentos é guiado pela demanda e, portanto, o planejamento começa com os pedidos de clientes disponíveis e previstos. Quanto maior o horizonte de planejamento, maior será a porcentagem de demanda prevista (STADTLER, 2005).

Os sistemas avançados de planejamento e programação da produção realizam o planejamento da demanda em três etapas (WAGNER, 2000): Primeiro, a previsão estatística identifica as principais características das séries temporais passadas e cria previsões usando análises de séries temporais e modelos causais (SILVER, PYKE e PETERSON, 1998). Em seguida, informações adicionais são incluídas na previsão, como ciclos de vida de grupos de produtos (WRIGHT e GOODWIN, 1998). Por fim, a colaboração entre diferentes áreas funcionais, como vendas, produção e compras, é apoiada para obter um consenso sobre a previsão (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.3.3. Planejamento de rede de suprimentos

O objetivo do planejamento da rede de suprimentos é uma utilização eficiente das capacidades de produção e armazenamento, determinadas pelo projeto da rede. O horizonte de planejamento deve cobrir pelo menos um ciclo sazonal, para equilibrar todos os picos de demanda, e é dividido em intervalos de tempo semanais ou mensais. Devido às mudanças nos dados de entrada, o planejamento da rede de abastecimento é atualizado de forma incremental, por exemplo, em semanas ou meses, para levar em consideração informações atualizadas, como perfis de estoque reais, usos de capacidade e novos dados de demanda. Por causa do tamanho do problema, é frequentemente necessário concentrar-se nos recursos gargalos (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

O planejamento da rede de abastecimento busca minimizar o custo de produção, transporte e manutenção de estoque, custo de atendimento atrasado ou

cancelado de pedidos de clientes e custo de capacidades adicionais. Restrições como prazos de entrega para pedidos de clientes, perecibilidade de produtos, capacidade de produção, armazenamento e transporte, calendários e limites inferior e superior em tamanhos de lote surgem nesse contexto. O planejamento da rede de abastecimento resolve os gargalos de capacidade produzindo em períodos anteriores ou posteriores, em locais alternativos ou em modos de produção alternativos, trabalhando horas extras, comprando produtos de fornecedores externos, ou atrasando ou cancelando a entrega (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

Se apenas uma planta for considerada, o planejamento da rede de abastecimento é semelhante ao planejamento mestre de produção clássico (*MPS*), com restrições de capacidade (NAHMIAS, 1997). Porém, certos recursos, como o cancelamento de entrega, não são considerados no *MPS* (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

O Planejamento da Rede de Fornecimento é baseado nas previsões semanais ou mensais por até um ano. Ele inclui pedidos fixos de clientes, os impactos de eventos de marketing de médio prazo e promoções nas vendas. Esses fatores são considerados na etapa de previsão estatística do planejamento da demanda. Além disso, a configuração da planta e o layout são determinados no módulo de design de rede e influenciam na estratégia de suprimentos. Os resultados deste planejamento são aplicados em outros módulos, como na programação detalhada da produção, que é baseada nas necessidades dos produtos finais e nas capacidades de produção (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.3.4. Cumprimento de demanda e *ATP*

O módulo de Atendimento à Demanda e Disponibilidade para Promessa (*Available-To-Promise - ATP*) é responsável por realizar a interface com os clientes e rastrear os pedidos dos clientes, desde a entrada até a entrega. Além disso, é nesse módulo que são considerados a promessa de pedido, a definição do prazo e o planejamento de escassez. A promessa de pedido é calculada a partir da combinação do estoque disponível e dos suprimentos previstos, com os pedidos já comprometidos pelos clientes. As quantidades restantes são as quantidades disponíveis para promessa (*ATP*), que podem ser usadas para prometer datas de entrega para novos

pedidos recebidos. Caso as quantidades *ATP* sejam insuficientes, os pedidos podem ser prometidos com base nas quantidades capazes de promessa (*Capable-To-Promise - CTP*), indicando a capacidade ociosa restante após a combinação da capacidade disponível com os pedidos dos clientes já comprometidos. Em caso de imprevistos, como a quebra de máquinas, entra em ação o planejamento de escassez, que especifica quais pedidos dos clientes comprometidos não serão atendidos a tempo (FISCHER, 2001; KILGER e SCHNEEWEISS, 2002; STADTLER, 2005).

2.3.5. Planejamento de distribuição e transporte

O Planejamento de Distribuição e Transporte refere-se ao cálculo de quantidades de distribuição (NEUMANN, SCHWINDT e TRAUTMANN, 2002). Aqui, é necessário gerenciar o transporte de mercadorias diretamente para os clientes e também através de armazéns e *cross docking*, com maior detalhamento do que no Planejamento Mestre. Quando os valores de produção não correspondem exatamente à demanda do período atual, regras e procedimentos são aplicados para direcionar o fluxo de mercadorias na cadeia de suprimentos. Por exemplo, em caso de escassez, o transporte de mercadorias será feito de forma que os estoques de destino de um item em diferentes centros de distribuição sejam preenchidos em igual proporção (STADTLER, 2005).

Com base nas ordens de produção a serem concluídas no dia seguinte ou turno, as cargas de caminhões para destinos diferentes são formadas. Isso também requer conhecimento detalhado dos pedidos pendentes dos depósitos e clientes, e deve levar em consideração as necessidades específicas dos clientes, como janelas de tempo para entrega e restrições legais para motoristas. O sequenciamento de localizações de clientes na viagem de um veículo é realizado por meio de modelos de roteamento de veículos (STADTLER, 2005).

2.3.6. Gestão de inventário

O Planejamento Mestre, bem como o Planejamento de Produção de Curto Prazo e a Programação Detalhada fornecem diretrizes para calcular as quantidades de aquisição a serem planejadas no módulo de Aquisição e Planejamento de

Necessidades de Material. Após desagregar os tipos de produtos ou famílias de produtos em itens, uma explosão da lista de materiais (*BOM*) é realizada para obter as quantidades necessárias de itens a serem adquiridos (STADTLER, 2005).

O Módulo de Gerenciamento de Estoque tem a função de receber as demandas de matéria-prima e gerenciar os tamanhos de lote adequados para compra (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.3.7. Programação detalhada de produção

A Programação Detalhada da Produção trata da alocação de recursos de curto prazo ao longo do tempo, para atender aos requisitos primários estabelecidos pelo Planejamento da Rede de Abastecimento. A tarefa é explodir os requisitos primários em lotes e programar esses lotes com recursos limitados. As necessidades secundárias de matérias-primas derivadas dos lotes determinados são, então, relatadas ao Gerenciamento de Estoque (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

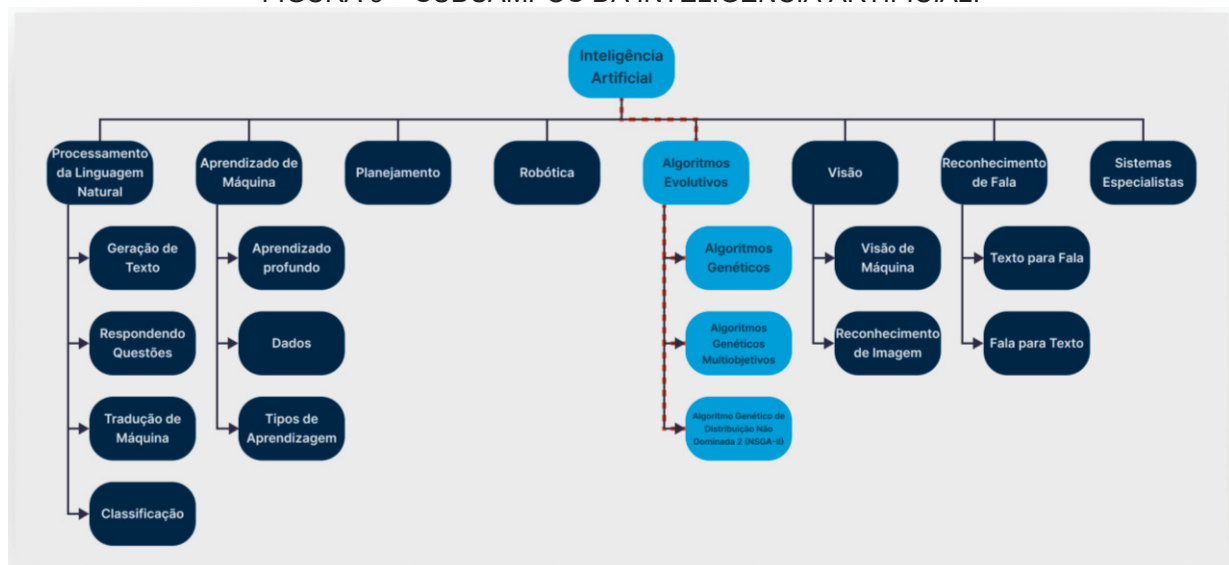
A Programação Detalhada da Produção é realizada individualmente para cada planta. O horizonte de planejamento corresponde ao tamanho do lote do Planejamento da Rede de Abastecimento. O objetivo é calcular um plano de produção viável que minimize uma função objetivo regulada, como a minimização do *makespan*, que permite respostas flexíveis às mudanças na demanda, especialmente se a planta inteira precisar ser reconfigurada antes de produzir outro conjunto de produtos finais. Outra função objetivo frequentemente considerada na prática é o atraso médio em relação às datas de vencimento para a entrega de produtos finais solicitados pelo cliente (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

Cada tarefa consome e produz diferentes produtos. No modo de produção em lote, a entrada de uma tarefa é consumida no início e a saída é obtida na sua conclusão. Alguns produtos intermediários podem ser armazenados em instalações de armazenamento com capacidade limitada. Outros produtos intermediários são perecíveis e não podem ser armazenados, devendo ser consumidos imediatamente. Nesse caso, as tarefas de produção devem ser atribuídas a tarefas de consumo, de modo que nenhum produto perecível esteja em estoque em nenhum momento (NEUMANN, SCHWINDT e TRAUTMANN, 2002).

2.4. ALGORITMOS INTELIGENTES

A Inteligência Artificial (IA) é uma área da ciência e engenharia que se concentra na teoria e prática do desenvolvimento de sistemas que apresentam características associadas à inteligência humana, como resolução de problemas, planejamento, aprendizado, adaptação, percepção, compreensão de linguagem e interação com o ambiente. O objetivo principal da IA é compreender os princípios que permitem o pensamento inteligente em humanos, animais e agentes artificiais, o que ajuda a abordar questões em diversas áreas de engenharia e outros campos de interesse. O desenvolvimento de sistemas com raciocínio automatizado é uma área multidisciplinar com grande potencial de desenvolvimento e aplicações, englobando a ciência da computação, engenharia, psicologia, neurociência, filosofia e outras áreas (TECUCI, p. 168, 2012).

FIGURA 9 – SUBCAMPOS DA INTELIGÊNCIA ARTIFICIAL.



Fonte: Adaptado de Kayid (2020) e Russel e Norvig (2020).

A IA oferece o potencial de solução de problemas em vários setores relevantes para a sociedade, incluindo negócios, manufatura, saúde, educação, militar e outros. Inovações recentes baseadas em IA, como reconhecimento facial e carros autônomos, podem ser desenvolvidas (WANG E SIAU, 2019). A IA tem vários subcampos, cada um com objetivos distintos, sendo os mais relevantes o Aprendizado de Máquina, Processamento de Linguagem Natural, Reconhecimento de Voz, Planejamento, Robótica, Visão (Reconhecimento de Imagens), Algoritmos Evolutivos (pertencente à

área de resolução de problemas), Sistemas Especialistas, entre outros (RUSSELL, NORVIG, 2021; KAYID, p. 2, 2020). Cada subcampo também tem suas áreas de estudo específicas, alguns destes podem ser encontrados na FIGURA 9.

2.4.1. Machine Learning

Traduzido do inglês como "*Machine Learning*", o aprendizado de máquina é descrito como "a tecnologia que permite que os computadores aprendam sem serem programados explicitamente. O sistema é capaz de aprender com dados, identificar padrões e tomar decisões sem interferência humana" (SAMUEL, 1959).

Os agentes de *Machine Learning* (ML) são sistemas computacionais que podem melhorar o seu comportamento através de uma análise aprofundada de sua experiência passada. Para isso, são necessários dados para construir modelos que possibilitem a realização de previsões e a solução de problemas através do uso de *softwares* (RUSSELL e NORVIG, p.59, 2021).

Como o ML depende dos dados fornecidos para funcionar, é importante categorizar os diferentes tipos de conjuntos de dados que existem em ambientes de trabalho. Estes podem ser divididos em Dados Estruturados, Dados Não Estruturados, Dados Semi Estruturados e Metadados, conforme QUADRO 2 (SARKER, p.3, 2021).

QUADRO 2 - TIPOS DE DADOS, DESCRIÇÃO E EXEMPLOS.

Tipos de Dados	Descrição	Exemplo
Estruturados	São dados organizados com uma estrutura bem definida, fáceis de acessar e que possuem relações entre si, geralmente dispostos em tabelas.	Nomes, datas, endereços, números de cartão de crédito e posições geográficas.
Não Estruturados	São dados sem formato predefinido, organização ou relações entre si, difíceis de detectar, processar e analisar. A maioria destes dados são textos e outros tipos de conteúdo de mídia.	Arquivos .pdf, e-mails, vídeos, áudios, imagens, apresentações e páginas da internet.
Semi Estruturados	São diferentes dos dados estruturados, pois não são armazenados em uma base de dados com relações entre as informações. No entanto, eles possuem uma certa organização que os torna mais fáceis de analisar do que os dados não estruturados.	HTML, XML, JSON e NoSQL
Metadados	Não são um tipo convencional de dados, mas sim "dados sobre dados". Eles são descrições informativas relevantes dos dados existentes, fornecendo um maior significado aos usuários.	O autor de um documento, seu tamanho ou palavras-chave

FONTE: Adaptado de Sarker (2021).

Na tecnologia de análise de dados e automatização da construção de modelos analíticos (aprendizado de máquina), existem diferentes tipos de abordagens para o aprendizado do sistema, incluindo Supervisionado, Não Supervisionado, Semi Supervisionado e por Reforço (SARKER, p. 4, 2021).

2.4.1.1. *Supervisionada*

O aprendizado de máquina é um processo no qual um programador fornece exemplos de entradas com resultados predefinidos para o sistema, utilizando dados rotulados. Esses dados são denominados "dados de treinamento", e consistem de pares de inputs e outputs. O algoritmo então aprende a identificar os rótulos ou valores desejados a partir de um novo conjunto de dados, sendo a assertividade do sistema avaliada e corrigida pelo programador, se necessário (SARKER, p. 4, 2021).

Dentro do aprendizado supervisionado, as técnicas de regressão e classificação são amplamente utilizadas para elaboração de modelos. A regressão é utilizada para previsões quantitativas e procura encontrar relações entre dados numéricos. É uma das técnicas pioneiras do aprendizado de máquina e constrói modelos preditivos a partir de equações variadas em complexidade (TALABIS *et al.*, 2014).

A classificação, por outro lado, é o processo de reconhecimento qualitativo de novos dados inseridos no sistema, categorizados de acordo com o treinamento realizado com base de dados estruturados ou não. O sistema prevê uma determinada categoria para os novos dados a partir de metodologias de programação, funções matemáticas e lógica. Existem três tipos de classificação: binária, multiclasse e multirrótulo (SARKER, p. 5, 2021).

A classificação binária se refere a tarefas de classificação com apenas duas possibilidades de resultado, como identificação de animais em imagens, verificação de e-mails como spam ou não, e detecção de câncer em pacientes (MONDAL, 2021). A classificação multiclasse, por outro lado, envolve mais de dois rótulos, mas não possui uma classificação normal ou anormal. A classificação multirrótulo também não possui classes com limites predefinidos, variando apenas de acordo com as capacidades computacionais (SARKER, p. 5, 2021).

2.4.1.2. Não supervisionada

Sem a necessidade de interferência humana, o aprendizado de máquina não supervisionado utiliza dados não rotulados com informações diversas, procurando encontrar uma correlação inédita entre resultados específicos. Ele realiza agrupamentos de dados (*data mining*), aplica curvas de densidade e realiza a redução dimensional de dados na estatística, a fim de encontrar tendências relevantes para o futuro. Com este caráter exploratório, há uma ampla gama de possibilidades para o modelo, permitindo obter conclusões e simulações rapidamente para projetos específicos, graças ao processamento da máquina (SARKER, 2021).

2.4.1.3. Semi supervisionada

Se trata de uma abordagem híbrida que une o método não supervisionado com o método supervisionado, pois opera com dados rotulados e não rotulados. Em aplicações práticas, há maior quantidade de dados não rotulados do que dados rotulados, tornando importante a utilização do modelo híbrido. Ao trabalhar com os dois tipos de dados, são mais eficientes as regressões e classificações realizadas em comparação com a abordagem supervisionada, e é possível também trabalhar com previsões e agrupamentos, como na metodologia não supervisionada. É utilizado em situações como traduções de máquina, detecção de fraude, rotulagem de dados e classificação de textos (SARKER, p.3, 2021).

2.4.1.4. Reforço

Neste caso, o algoritmo é condicionado a permitir que o *software* avalie um comportamento ótimo de acordo com o contexto ou ambiente no qual se encontra. Essa lógica de programação funciona por meio de um sistema de penalidades e recompensas, tendo o objetivo de maximizar as recompensas e minimizar os riscos de punição, por meio de suas ações. É uma ferramenta importante para treinar modelos de IA que auxiliam na automação, robótica, condução automática de veículos e solução de problemas de logística de produção. No entanto, não é uma abordagem recomendada para solucionar problemas conhecidos e diretos (SARKER, p.3, 2021).

2.4.2. Algoritmos evolutivos

Algoritmos Evolutivos (AEs) são definidos como variantes de métodos de solução estocástica e otimização heurística, utilizados quando não existe uma solução analítica conhecida ou não é possível elaborá-la. Os modelos utilizados para os AEs são baseados na teoria de seleção natural, onde existe uma população de indivíduos, ou possíveis soluções, dos quais as mais aptas são selecionadas para a realização de um cruzamento ou combinação, onde mais soluções possíveis são geradas. Dos indivíduos remanescentes, podem ser realizadas mais combinações e mutações que serão avaliadas e filtradas de acordo com a demanda de cada problema. Existe uma ampla gama de AEs que variam de acordo com o tamanho da população existente e com o tipo de representação dos indivíduos (RUSSELL e NORVIG, 2021).

Já Silva *et al.* (2018) definem que os algoritmos evolutivos (AEs) são técnicas computacionais de otimização baseadas em processos de evolução biológica, que utilizam seleção natural, recombinação e mutação para encontrar soluções para problemas complexos. Esses algoritmos são aplicáveis em diversas áreas do conhecimento, como engenharia, finanças, biologia, entre outras.

Um exemplo de algoritmo evolutivo é o algoritmo genético, que simula o processo de seleção natural, mutação e recombinação para encontrar soluções ótimas em problemas de otimização. Esse algoritmo utiliza uma população de indivíduos, que são representações das possíveis soluções do problema, e através de um processo de seleção natural, os indivíduos mais aptos são selecionados para a próxima geração, enquanto os menos aptos são eliminados (GOLDBERG, 1989). O algoritmo genético tem sido amplamente utilizado em diversas áreas, como otimização de sistemas de controle, redes neurais, processamento de sinais, entre outras (BÄCK *et al.*, 2000).

Outro exemplo de algoritmo evolutivo é o algoritmo de evolução diferencial, que se baseia em um processo de recombinação de vetores para gerar novas soluções. Nesse algoritmo, são criados vetores aleatórios para cada indivíduo da população, que são combinados por meio de um processo de recombinação para gerar novos vetores, que são então avaliados e selecionados para a próxima geração (STORN e PRICE, 1997). O algoritmo de evolução diferencial tem sido aplicado em diversas áreas, como reconhecimento de padrões, processamento de sinais e otimização de sistemas (QIN *et al.*, 2009).

Além dos algoritmos genéticos e de evolução diferencial, existem diversas outras técnicas de algoritmos evolutivos, como programação evolutiva, estratégias evolutivas, algoritmos de colônia de formigas, entre outras (SILVA *et al.*, 2018). Todos esses algoritmos têm em comum o uso de processos de evolução biológica para encontrar soluções para problemas complexos, e cada um tem sua própria aplicação específica.

2.4.2.1. Algoritmo genético

O Algoritmo Genético (AG) é um método de busca evolutiva e estocástico que pode fornecer soluções ótimas ou próximas da ótima para problemas de otimização combinatória. Ele se baseia na mecânica de população e seleção. As soluções em potencial são codificadas como "genes" e novas soluções podem ser criadas por meio de "mutação" entre os membros da população atual ou "acasalamento" entre duas soluções. As melhores soluções são selecionadas para se reproduzir, enquanto as piores são descartadas. Os algoritmos genéticos são também métodos de busca probabilísticos, o que significa que os estados que eles exploram não são determinados exclusivamente pelas propriedades dos problemas. Eles são usados na área da inteligência artificial, assim como outros algoritmos de busca, para explorar soluções potenciais com o objetivo de encontrar uma solução que resolva o problema. Eles têm aplicações em vários campos, incluindo engenharia, biologia, ciência da computação e ciências sociais. Uma das características mais atrativas dos AGs é sua flexibilidade em lidar com várias funções objetivo sem necessidade de propriedades matemáticas complexas (GEN e CHENG, 1997; SHAPIRO, 1999; LEE, JEONG e MOON, 2002).

Os principais fatores na implementação de um AG incluem a representação cromossômica, a inicialização da população, uma medida de avaliação, cruzamento, mutação e estratégia de seleção. Além disso, parâmetros genéticos, como tamanho da população (*pop_size*), número de gerações (*max_gen*), probabilidade de cruzamento (*Pc*) e probabilidade de mutação (*Pm*), precisam ser determinados antes de executar o AG (LEE, JEONG e MOON, 2002).

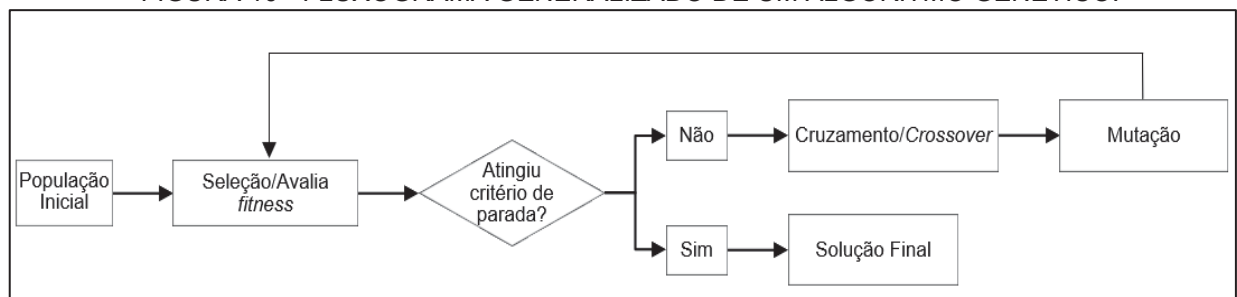
A representação dos Algoritmos Genéticos (AGs) é feita através de sequências de caracteres retiradas de algum alfabeto $A = A1, A2, \dots, AL$. Essas sequências são conhecidas como cromossomos e os componentes deles, $A1, A2, \dots$, são chamados

de genes. Os valores que cada gene pode tomar são conhecidos como alelos. No caso de cromossomos binários, os alelos são *0* ou *1*. Por exemplo, o cromossomo $A = 11110000$ significa que os quatro primeiros genes possuem o alelo *1* e os últimos quatro possuem o alelo *0* (SHAPIRO, 1999).

A função objetiva do algoritmo é encontrar cromossomos excelentes, ou seja, a aptidão é a medida da excelência de um cromossomo. A aptidão é determinada por uma função de aptidão, que é uma função monótona da função objetiva. Essa função determina como os cromossomos serão multiplicados ou descartados durante a evolução do algoritmo. A função objetiva e a função de aptidão são frequentemente usadas de forma intercambiável na comunidade AG, mas é importante separá-las. Na terminologia biológica, a aptidão é o aumento devido à seleção, enquanto a função objetiva é uma propriedade do problema a ser resolvido (SHAPIRO, 1999).

Um algoritmo genético funciona em uma população de cromossomos. Começa com uma população aleatória e evolui para uma população de cromossomos excelentes. O algoritmo funciona com base em operações que produzem a população na próxima geração, levando à melhora da aptidão geral da população. Essas operações são repetidas até que um critério de parada, como um determinado número de cromossomos ou uma série de qualidade, seja atingido (SHAPIRO, 1999).

FIGURA 10 - FLUXOGRAMA GENERALIZADO DE UM ALGORITMO GENÉTICO.



Fonte: Adaptado de Shiki, Franzoni, da Silva e Brito (2010).

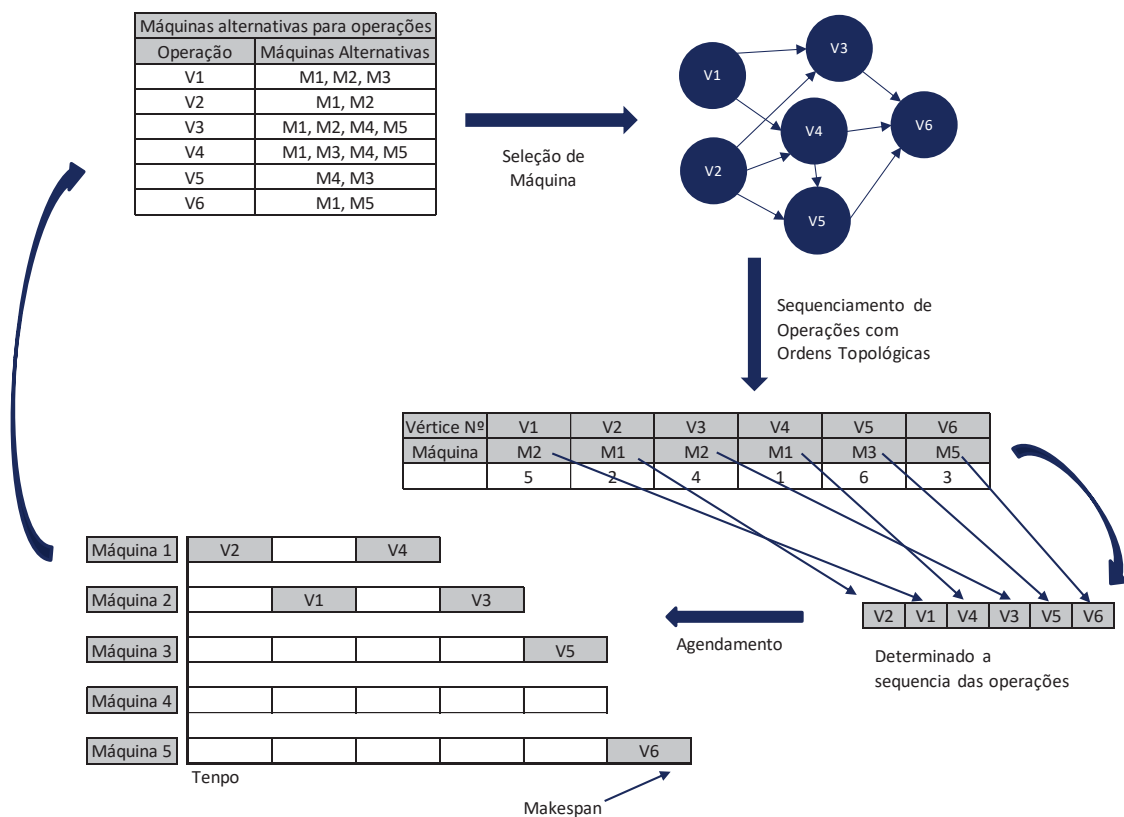
O Algoritmo Genético Simples (AG) funciona com uma população constante de tamanho N . Três operações levam a população na geração t para produzir a nova população na geração $t + 1$: seleção, cruzamento (recombinação) e mutação (SHAPIRO, 1999) conforme FIGURA 10.

- A Seleção escolhe sequências de cromossomos da população atual de forma que as sequências melhores tenham uma maior probabilidade de serem escolhidas. A seleção não adiciona novas sequências. Ao ser

aplicada repetidamente, ela apenas reorganiza os cromossomos da população, aumentando o número de sequências adequadas e diminuindo o número de sequências menos adequadas.

- O Cruzamento (recombinação) produz filhos de um par de pais, combinando partes de cada pai. Cada descendente tem partes dos cromossomos de cada pai.
- A Mutação altera aleatoriamente caracteres em uma sequência de filhos. Cada caractere é alterado com uma pequena probabilidade.

FIGURA 11 – REPRESENTAÇÃO DO CROMOSSOMO.



FONTE: Adaptado de Lee, Jeong e Moon (2002).

Em um problema de planejamento usando AG, é fundamental desenvolver uma representação adequada para representar uma solução viável. Representar um caminho com restrições de precedência em um gráfico pode ser difícil. Para gerar ordens topológicos, a representação deve ser capaz de gerar todas as ordens topológicas possíveis para uma dada atividade na rede de vértices. Qualquer rota da solução sempre corresponde a ordens topológicas. Suponha que haja um trabalho com seis operações, $v1$ a $v6$. A estrutura do cromossomo pode ser representada como

mostrado na FIGURA 11 – REPRESENTAÇÃO DO CROMOSSOMO. FIGURA 11 (LEE, JEONG, e MOON, 2002).

Na FIGURA 11, a primeira linha de um cromossomo indica as operações que combinam um número de máquina selecionado aleatoriamente para cada operação. Cada operação seleciona um número de máquina aleatoriamente entre as possíveis alternativas. A segunda linha indica a prioridade para a seleção de candidatos quando há operações sem arestas de entrada. O valor de um gene é gerado exclusivamente em $[1, J]$, onde J é o número total de operações. Por exemplo, se $v1$ e $v2$ não têm restrições de precedência, $v2$ é selecionado para a sequência de operação porque o número de prioridade de $v2$ é 2. Depois de selecionar $v2$, podemos remover $v2$ e todas as arestas que conduzem a ele. Agora, $v1$ e $v5$ não têm predecessores, e $v1$ é selecionado para a próxima operação porque tem prioridade mais alta do que $v5$. Da mesma forma, a sequência de operação $v2-v1-v4-v3-v5-v6$ é determinada para um trabalho e as máquinas $M1-M2-M1-M2-M3-M5$ são usadas para cada operação na ordem. (LEE, JEONG, e MOON, 2002).

Por exemplo, suponha que a máquina alternativa e o tempo de processamento para cada operação são os apresentados na TABELA 1.

TABELA 1 – MÁQUINAS ALTERNATIVAS E TEMPOS POR PROCESSOS PARA OPERAÇÃO VJ.

Máquina Nº	Operação					
	V1	V2	V3	V4	V5	V6
1	5	5	6	3	2	2
2	5	5	4	∞	∞	∞
3	6	∞	6	5	3	∞
4	∞	∞	∞	3	∞	∞
5	∞	∞	6	4	∞	3

FONTE: Adaptado de Lee, Jeong e Moon (2002).

O tempo de início de $v2$ na máquina $M1$ é zero, e o tempo de término de $v2$ é 5. Para a próxima operação, $v1$ na máquina $M2$, o tempo de início deve ser igual ao tempo de término de $v2$ e o tempo de término é 10 (tempo de término de $v2$ + tempo de processamento de $v1$). Continuando nesse processo, chegamos ao resultado apresentado na TABELA 2 (LEE, JEONG e MOON, 2002).

A TABELA 2 pode ser representada em um gráfico de *Gantt*, como mostrado na FIGURA 11, e o *makespan* é 23. Quando os cromossomos são combinados e modificados por operações genéticas, como cruzamento ou mutação, o *makespan*

pode ser reduzido. Quando o *makespan* é minimizado o suficiente, o cromossomo nos apresenta a melhor seleção de máquina e sequência de operações (LEE, JEONG e MOON, 2002).

TABELA 2 – HORÁRIO DE INÍCIO E TÉRMINO DE CADA OPERAÇÃO (HORA DE INÍCIO, HORA DE TÉRMINO).

Máquina Nº	Operação					
	V1	V2	V3	V4	V5	V6
				10,13		
1		0,50				
2	5,10		13,17			
3					17,20	
4						
5						20,23

FONTE: Adaptado de Lee, Jeong e Moon (2002).

Desta forma, a primeira etapa do Algoritmo Genético (AG) é inicializar a população de cromossomos. Cada cromossomo contém uma prioridade e uma máquina selecionada aleatoriamente do conjunto de máquinas alternativas para cada operação. O processo de inicialização é executado seguindo o procedimento apresentado na FIGURA 12 e gera uma população de cromossomos (LEE, JEONG e MOON, 2002).

FIGURA 12 – GERAÇÃO DE SOLUÇÃO VIÁVEL.

```

Procedure: Feasible solution representation
Input data:
  Number of parts and alternative machines
  Number of operations for each part;
  Let  $v_j$  ( $j=1, \dots, J$ ) be the  $J$  operations for all part types in an orderly manner;
  Set of directed graphs with precedence constraints for all parts;
  Set of alternative machines and time for each operation including outsourcing;
  Transportation time for outsourcing;
Begin:
  Generate Chromosome with Machine selection:
    Allocate a randomly selected machine from the set of alternative machines
    for each operation to  $v_j$ ;
    Generate a random priority number (1 to  $J$ ) for each operation,  $v_j$ ;

  Generate operation-sequences:
    While (any vertex remains) do
      If every vertex has a predecessor then
        the network is infeasible: stop;
      Else pick a vertex  $v_j$  which has no predecessors;
        Output  $v_j$ ;
        Delete  $v_j$  and all edges leading from  $v_j$  from the directed graph;
      End;
    Generate a feasible schedule:
      Make a schedule with selected machines considering the operation-sequence for
      all parts;
End;

```

FONTE: Lee, Jeong e Moon (2002).

Os cromossomos contêm muitas informações. Eles alocam operações para cada máquina de maneira limitada por determinadas sequências de operações. A avaliação da aptidão é mostrada na FIGURA 13 (LEE, JEONG e MOON, 2002).

FIGURA 13 – PROCEDIMENTO DE AVALIAÇÃO DE APTIDÃO.

```

Procedure: Fitness evaluation
Begin:
   $MP_{rk} = 0$  for all  $k$ ;
   $S_{ijk} = 0$  for all  $i, j, k$ ;
  Determine the sum of process time using machine  $k$ ;
  For  $r = 1$  to  $R$ 
    If  $r > 1$  then  $MP_{rk} = MP_{(r-1),k}$ ;
    Else;
      For  $i = 1$  to  $I_r$  do
        For  $j = 1$  to  $J_i$  do
           $k?$  selected machine number for operation  $j$ ;
          If  $k$  is the outsourcing machine then;
            If operation  $j$  is the first operation then;
               $S_{rjk} = MP_{rk}$ ;
            Else if operation  $j-1$  did not use machine  $k$  then;
               $S_{rjk} = TR_k + MP_{rk}$ ;
            Else  $S_{rjk} = MP_{rk}$ ;
               $E_{rjk} = S_{rjk} + mat_{rjk}$ ;
          If operation  $j+1$  will not use machine  $k$  then;
             $MP_{rk} = E_{rjk} + TR_k$ ;
          Else  $MP_{rk} = E_{rjk}$ ;
        Else;
           $S_{rjk} = MP_{rk}$ ;
           $E_{rjk} = S_{rjk} + mat_{rjk}$ ;
           $MP_{rk} = E_{rjk}$ ;
      End;
    End;
  End;
  For  $r = 1$  to  $R$  do
     $MS_r = \max_{k=1, \dots, K'} (MP_{rk})$ ;
  End;
  Return  $MS$ ;
End;

```

FONTE: Lee, Jeong e Moon (2002).

A função de adequação do modelo APS leva em conta apenas um objetivo: minimizar o *makespan* (MS_r) para cada ordem r . O *makespan* é o tempo necessário para concluir todas as operações de todas as peças. Assim, o tempo de conclusão da última operação é o tempo de *makespan*, desde que esteja limitado por uma data de vencimento. Para todas as ordens r , MS_r deve satisfazer a equação $MS_r \leq Dr - ATr - DTr$ (LEE, JEONG e MOON, 2002).

Após a avaliação da adequação de todos os cromossomos gerados, a estratégia de seleção escolhe os cromossomos da população. Em seguida, uma nova população é criada para a próxima geração com base nos pais e filhos, usando o método de seleção conhecido por roda de roleta (GEN e CHENG, 1997; LEE, JEONG e MOON, 2002).

FIGURA 14 – ILUSTRAÇÃO DE CROSSOVER BASEADO EM PEDIDO.

(1) Seleciona aleatoriamente uma substring do pai 1

		Parte 1		Substring selecionada - Parte 2		Parte 3
Pai 1	V1	V2	V3	V4	V5	V6
	M2	M1	M2	M1	M3	M5
	5	1	2	4	6	3

Pai 2	V1	V2	V3	V4	V5	V6
	M5	M3	M1	M2	M1	M2
	2	6	1	4	3	5

(2) Produz um pai 1 modificado.

Pai 1	V1	V2	V3	V4	V5	V6
	M2	M1	M2	M1	M3	M5
	5	1	2	4	6	3

Filho	V1	V2	V3	V4	V5	V6
	M5	M3	M2	M1	M1	M2
	6	1	2	4	3	5

Pai 2	V1	V2	V3	V4	V5	V6
	M5	M3	M1	M2	M1	M2
	2	6	1	4	3	5

FONTE: Adaptado de Lee, Jeong e Moon (2002).

Para produzir a próxima geração, os operadores genéticos (seleção, cruzamento e mutação) são executados, produzindo um novo conjunto de cromossomos. O cruzamento (*crossover*) é o principal operador e exerce uma grande influência no desempenho da abordagem AG, enquanto a mutação é um operador secundário. O documento usa cruzamentos baseados em ordem, onde a ordem das tarefas de um pai é imposta às tarefas correspondentes no outro pai. Já as mutações baseadas em ordem trocam aleatoriamente as posições dos números de prioridade. Um exemplo de um *crossover* baseado em ordem é apresentado na FIGURA 14 (LEE, JEONG e MOON, 2002).

Suponha que existam dois cromossomos que são pais com números de máquina [2 1 2 1 3 5] e [5 3 1 2 1 2]. As colunas pares da esquerda representam, respectivamente, a parte 1, parte 2 e parte 3. Utilizando o mesmo procedimento, é possível produzir descendentes [6 1 2 4 3 5] com números de máquina [5 3 2 1 1 2]. O operador de mutação "swap" é apresentado neste exemplo. Este operador seleciona

aleatoriamente dois genes (números de prioridade) dentro de um cromossomo e troca seus conteúdos (LEE, JEONG, e MOON, 2002).

Denomine $P(t)$ e $C(t)$ como pais e filhos na geração atual t . O procedimento geral da abordagem baseada em Algoritmo Genético proposta é descrito na FIGURA 15 (LEE, JEONG, e MOON, 2002).

FIGURA 15 – PROCEDIMENTO GERAL DA ABORDAGEM BASEADA EM AG.

```

Procedure: GA-based approach
Begin:
     $t \leftarrow 0$ ;
    initialize parent population  $P(t)$ ;
    evaluate  $P(t)$ ;
    While (not termination condition) do
        recombine  $P(t)$  to  $C(t)$  by crossover and mutation;
        evaluate  $C(t)$ ;
        select  $P(t+1)$  from  $P(t)$  and  $C(t)$ ;
         $t \leftarrow t + 1$ ;
    End;
End;

```

FONTE: Lee, Jeong e Moon (2002).

2.4.2.2. Algoritmos genéticos de múltiplos objetivos

Conforme os algoritmos genéticos foram evoluindo de acordo com as necessidades de variados campos de aplicação, algoritmos que possuem como alvo mais de um objetivo passaram a ser explorados e testados.

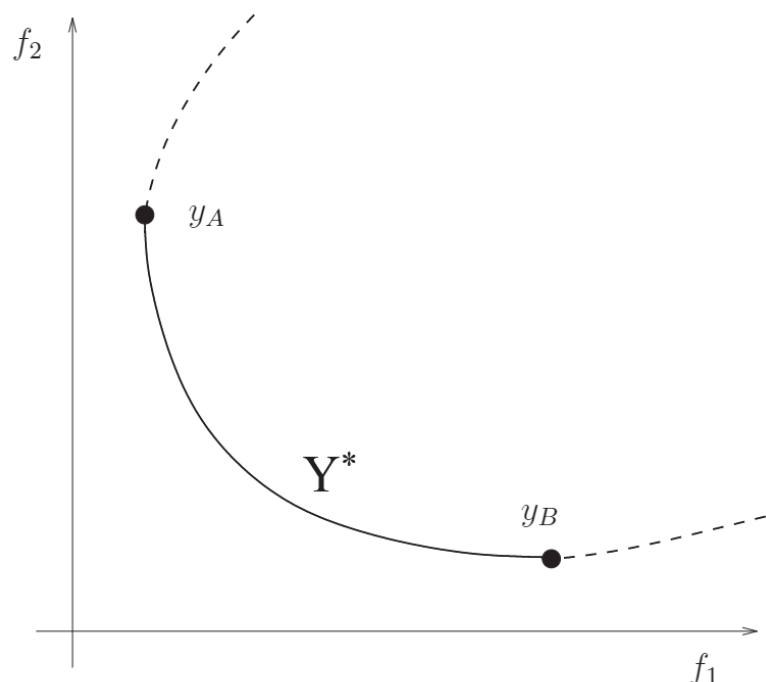
Por se tratar de uma otimização simultânea de duas funções que podem ser conflitantes ou não, algoritmos multiobjetivo possuem duas abordagens principais para a representação dos resultados finais da função alvo. A primeira, seria realizar a combinação das funções objetivo em uma única função objetivo através de metodologias como a teoria da utilidade e o método da soma ponderada. Essas técnicas, no entanto, trazem complicações devido à necessidade de definir precisamente os pesos corretos para cada objetivo, de maneira que alterações mínimas podem consideravelmente alterar a resposta final do algoritmo (KONAK, COIT e SMITH, 2006).

Na segunda abordagem, conforme Konak, Coit e Smith (2006), a resposta final do algoritmo é representada por um conjunto com as melhores soluções de acordo com seus respectivos resultados para cada função objetivo. Nesse caso, dentro do conjunto de soluções final obtido após n iterações nenhuma solução é necessariamente superior

ou dominante perante a outra, sempre existindo algum tipo de troca (*trade off*) ao comparar os valores da função alvo de uma solução com a outra. Essa abordagem é preferível para a tomada de decisões por permitir a escolha da melhor solução do conjunto conforme a relevância de cada função fitness para determinado problema a ser explorado. Em adicional, a abordagem também elimina a necessidade de atribuir pesos de proporção arbitrária que podem trazer uma variabilidade indesejada no final de cada iteração do algoritmo.

Para algoritmos genéticos multiobjetivos onde a resposta final abrange múltiplas soluções, o objetivo é que o conjunto de soluções convirja para o mais próximo possível do conjunto de soluções ideais de Pareto, que compõem a fronteira de Pareto. Entre essas soluções, deve-se ter como objetivo também que determinada diversidade seja mantida, de modo a ampliar o espaço de busca que permite que novas soluções sejam exploradas conforme ocorrem as iterações (DEB *et al.*, 2002).

FIGURA 16 – FRONTEIRA DE PARETO, COM EXTREMIDADES DEFINIDAS POR y_A E y_B , ONDE AS MÍNIMAS PARA CADA FUNÇÃO SÃO OBTIDAS.



Fonte: Modificado de Takahashi (2007).

A fronteira de Pareto é um grupo de resultados considerados ideais que não podem ser superados quando dois objetivos estão sendo considerados. Por exemplo, ao realizar uma otimização com o intuito de obter resultados para minimizar duas funções objetivo $f_1(x_1)$ e $f_2(x_2)$, existe um grupo de resultados Y^* com uma variedade

de pares $x1$ e $x2$ que atendem ao máximo essa premissa de otimização, não sendo necessariamente dominados por quaisquer outros pares fora do grupo (TAKAHASHI, 2007).

Na FIGURA 16, observam-se os pares extremos apresentados como yA ($x1A$, $x2A$) e yB ($x1B$, $x2B$), onde yA representa o par onde é obtido o menor valor para $f1(x1)$ e yB onde é obtida o menor valor para $f2(x2)$. Dentre todos os resultados do grupo, nota-se que existe um trade off na medida que se prioriza a minimização de uma função sobre a outra, de maneira que yA ao resultar no melhor e menor valor para $f1$ também resulta no maior e pior valor para $f2$ do conjunto.

Deb *et al.* (2002) propuseram um algoritmo genético de múltiplo objetivo denominado NSGA-II que permite que as soluções convirjam com maior velocidade ao mais próximo possível da fronteira de Pareto em comparação aos outros algoritmos evolutivos como o PAES e SPEA. O algoritmo em questão possui caráter elitista e também elimina a necessidade de definir uma variável de compartilhamento para manter a diversidade da solução, o que é necessário para os outros algoritmos citados onde o grau de complexidade de funcionamento é intensificado.

2.5. PROPOSIÇÕES

Nesta revisão da literatura, é possível perceber os avanços dos sistemas de informação aplicados à manufatura desde meados da década de 1970. Esses avanços foram originados pela explosão da *Bill of Material*, com os sistemas *MRP*. Em seguida, houve um interesse por recursos materiais e máquinas produtivas, resultando no *MRP II* e *CRP*.

Com o crescimento dos sistemas para a manufatura, surgiram os sistemas *ERP*, que tinham um foco mais gerencial e na administração da empresa. No entanto, os problemas de planejamento persistiram nas fábricas, o que deu oportunidade para o surgimento dos primeiros sistemas de *FCS*, que se concentraram na capacidade finita e nas restrições do chão de fábrica, resultando na aparição dos primeiros sistemas de *APS*.

O foco do APS é resolver problemas de planejamento, como data de entrega, capacidade, gargalos de recursos, data de vencimento, etc. Ao longo dos anos, surgiram várias modelagens matemáticas, abordagens e metodologias para solucionar

estes problemas de *APS*. No entanto, ainda há pouco enfoque na aplicação de algoritmos de inteligência artificial para o problema de *APS*.

Nesse sentido, os algoritmos genéticos são uma das abordagens de inteligência artificial que podem trazer avanços significativos para os sistemas de *APS*. Esses algoritmos são uma técnica de otimização que utiliza conceitos de evolução biológica, como seleção natural, recombinação e mutação, para encontrar soluções ótimas para problemas complexos. O algoritmo genético multiobjetivo, por exemplo, é capaz de lidar com problemas que envolvem múltiplos objetivos, enquanto o algoritmo genético de classificação não dominada 2 (NSGA II) é capaz de gerar soluções ótimas em um espaço de solução não dominado.

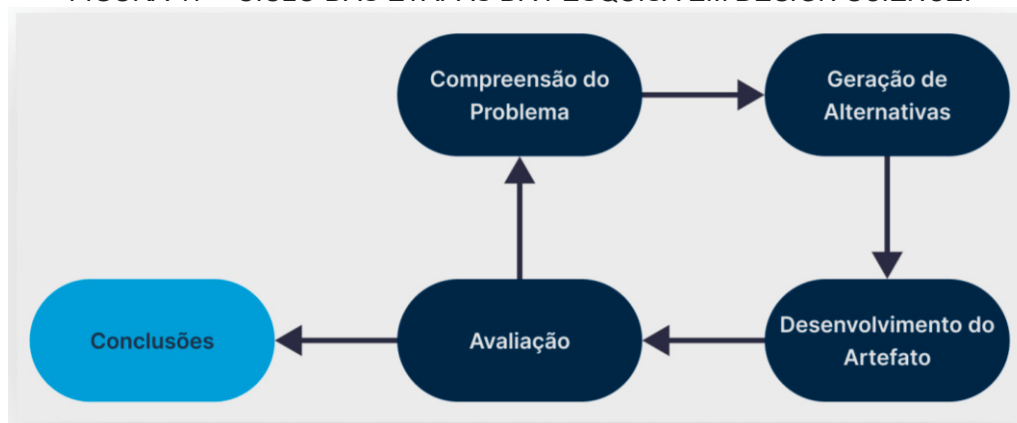
Desta forma, algoritmos que possam analisar dados e sugerir cenários de programação da produção, além de aprender com a base histórica, podem trazer avanços significativos para os sistemas de *APS*, sendo possível que sejam a próxima geração dos sistemas de planejamento de produção aplicados à manufatura.

3. ABORDAGEM METODOLÓGICA

Para testar uma plataforma adequada para desenvolver o APS, aprovar o artefato, obter informações sobre a utilização do sistema APS e adquirir conhecimento acadêmico através de pesquisa científica, a abordagem metodológica escolhida foi o *Design Science Research (DSR)*. Este método consiste em aprender através da construção e avaliação do artefato, cujo objetivo é resolver um ou uma cadeia de problemas, sem produzir um produto final, mas apenas o artefato (LACERDA *et al.*, 2013). A característica construtiva e prospectiva do *DSR* busca estabelecer o "como deveria ser", em contraste com a característica analítica de outros métodos que buscam entender o "como é" do mundo real (SANTOS, 2018). Um artefato é definido por Simon (1996 *apud* DRESCH *et al.*, 2015) como a organização dos componentes do ambiente interno para alcançar objetivos em um determinado ambiente externo.

Este método foi escolhido porque, de acordo com Santos (2018), é adequado em um projeto de pesquisa que visa criar um artefato para melhorar o mundo real presente ou futuro, aplicado em um contexto de cooperação ou não com os atores envolvidos. A efetividade do artefato em alcançar as melhorias é o foco do estudo.

FIGURA 17 – CICLO DAS ETAPAS DA PESQUISA EM *DESIGN SCIENCE*.



FONTE: Adaptado de Santos (2018).

O processo de realização de uma pesquisa em *Design Science*, segundo Santos (2018), tem naturalmente semelhanças diretas com o processo de *Design*. As fases desta metodologia (FIGURA 17), originalmente determinadas por Manson (2006), estão abertas à customização de acordo com as necessidades do experimento, e são as seguintes: a) Compreensão do problema; b) Geração de Alternativas; c) Desenvolvimento do Artefato; d) Avaliação; e) Conclusões.

Este processo começa com a compreensão do problema, seja parcial ou completa. Na *Design Science*, compreensões parciais são possíveis, pois o artefato em si pode ajudar a aprimorar a definição do problema e permitir um novo ciclo de geração de alternativas e desenvolvimento do artefato. O pesquisador deve refletir sobre a relação entre o problema em questão e problemas similares encontrados em outros contextos ou problemas mais amplos, para os quais os resultados do estudo seriam relevantes. A compreensão do problema pode ser facilitada por meio de uma Revisão Bibliográfica Sistemática (RBS) e Revisão Bibliográfica Assistemática (RBA), bem como por outros métodos de pesquisa, como Etnografia, Estudo de Caso ou uma Survey (SANTOS, 2018). De acordo com Manson (2006), o objetivo principal é identificar qual é o problema. Em seguida, é crucial definir onde o problema ocorre para determinar o escopo do artefato. Depois, é necessário estabelecer métricas ou parâmetros de aceitação para solução do problema, como quanto o artefato precisa fazer para alcançar a solução desejada. Por fim, é importante considerar o problema da cadeia, que é definido como outros problemas organizacionais que podem ter um artefato específico ou não para mitigá-los ou resolvê-los, e o artefato da pesquisa pode ter um impacto sobre esses problemas (LACERDA *et al.*, 2013).

Sugerir uma solução não é uma tarefa fácil. Durante a fase de geração de alternativas, é realizado um levantamento de como esse artefato pode ser construído do zero ou adaptado de algo existente. É feita uma listagem de tudo o que é necessário durante todo o processo e são anotadas todas as falhas e acertos desde as primeiras etapas até o final. Nesta fase, também são exigidas verificações em questões de conformidade e regulamentações entre tecnologia, leis e/ou direitos humanos quando aplicadas (MANSON, 2006). Santos (2018) define esta fase como um processo essencialmente criativo, que pode ser realizado de forma solitária pelo pesquisador ou de forma colaborativa envolvendo outros pesquisadores e os próprios atores relacionados ao problema ou à solução. Técnicas convencionais de criação utilizadas pelo *Design*, como 635 e *Brainstorming*, podem ser adotadas nesta etapa, assim como técnicas de representação de alternativas, como *storyboards*, *mock-ups* e infográficos.

Na etapa de desenvolvimento, o artefato real ganha vida. Nesta fase é explicado por que a ferramenta atual foi escolhida para este projeto e como foi construído esse artefato específico, além disso, há uma explicação dos componentes do artefato e como eles podem ser usados (MANSON, 2006). Este desenvolvimento

pode envolver diferentes abordagens, como infográficos, algoritmos computacionais, *mock-ups*, modelos, maquetes e protótipos com diferentes níveis de fidelidade (LACERDA *et al.*, 2013).

A etapa de avaliação consiste em explicar como o artefato será avaliado e como será alcançado o que se espera. É importante criar um método de teste para o artefato antes de ativá-lo. Compare os resultados obtidos com os resultados esperados e avalie a necessidade de mudanças, se necessário (MANSON, 2006). O método da *Design Science* busca tanto a validade científica (rigor na concepção e condução da pesquisa) quanto a validade pragmática (eficácia e efetividade das soluções). A meta principal da etapa de avaliação é confrontar a proposição prescritiva, consubstanciada no artefato, com sua efetividade no mundo real (SANTOS, 2018).

Uma vez concluído o(s) ciclo(s) de avaliação(ões), são realizadas as considerações finais e conclusões do estudo. É recomendado que as conclusões estejam alinhadas ao problema central da pesquisa e seus objetivos, sintetizando de forma textual ou visual as lições aprendidas durante as fases do projeto, e destacando as contribuições deste trabalho para a Classe de Problemas identificada na etapa de Compreensão do Problema (SANTOS, 2018).

Para o desenvolvimento de um *framework* de *software APS* com algoritmos inteligentes, objeto deste estudo, foram seguidas as 5 etapas do *DSR*, como sugerido por Manson (2006).

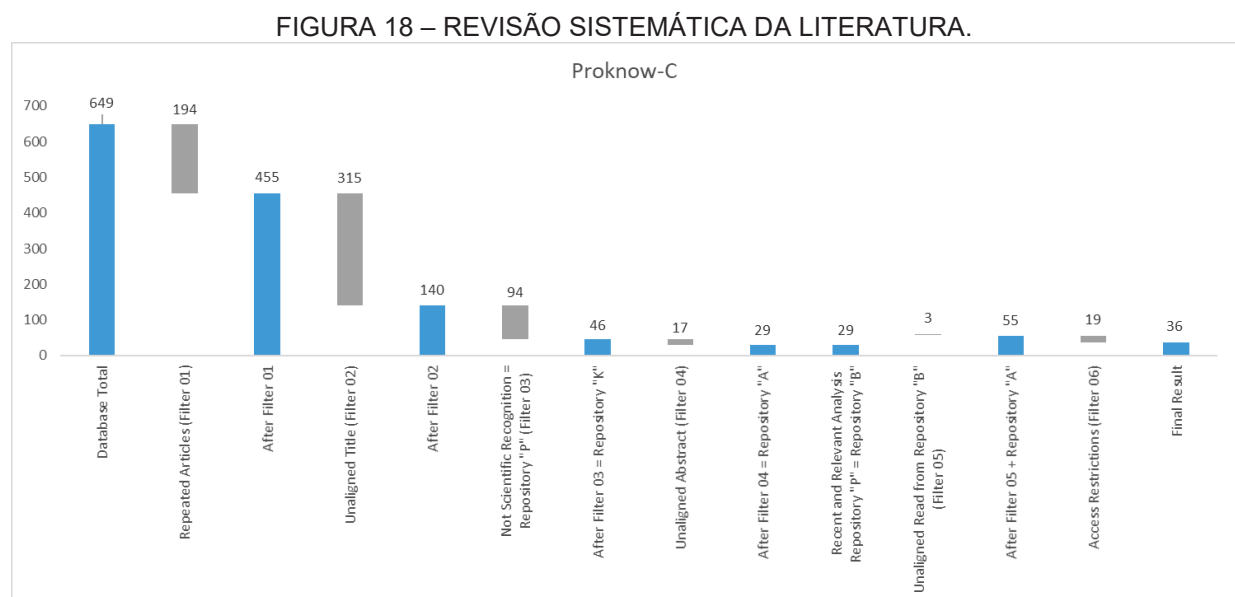
4. DESENVOLVIMENTO

A metodologia *DSR* é uma abordagem que busca desenvolver soluções inovadoras para problemas práticos, com base em princípios teóricos, a partir de uma compreensão profunda do contexto em que o problema está inserido. Ela envolve a criação de artefatos, que são soluções concretas para problemas específicos, e a avaliação desses artefatos, a fim de garantir que eles atendam aos requisitos dos usuários e do ambiente em que serão aplicados.

Desta forma, a seguir serão apresentados os detalhes do processo de desenvolvimento da solução *OPTimizIA*, por meio da metodologia *DSR*, desde a análise bibliográfica sistemática até a avaliação dos resultados obtidos. Serão descritas as etapas de criação dos artefatos, como o algoritmo genético multiobjetivo *NSGA-II* e a plataforma web de programação da produção, bem como as técnicas utilizadas para avaliar a solução proposta.

4.1. ETAPA 1 – COMPREENSÃO DO PROBLEMA

A primeira fase da pesquisa tem como objetivo determinar o problema. Para tal, foi utilizada a metodologia de Revisão Bibliográfica Sistemática (RBS).



FONTE: Adaptado de Vieira, Deschamps e Valle (2021).

Com a realização da RBS, como já foi amplamente explorada na seção 1.2 deste documento e com seus resultados mostrados na FIGURA 18, é possível afirmar

que o principal problema é a falta de evolução dos sistemas avançados de planejamento da produção (APS) no contexto da Indústria 4.0. A cadeia de problemas foi determinada como dependência humana, inflexibilidade, uso de planilhas, falta de integração com produtos *IoT* e *RFID*, e falta de uso de algoritmos inteligentes, todos abordados por Vieira, Deschamps e Valle (2021).

Neste contexto e com base nos problemas identificados na revisão bibliográfica sistemática foi definido o problema da pesquisa, já apresentado anteriormente, que é: “Como é possível desenvolver um *framework* de um sistema avançado de planejamento e programação da produção que utilize algoritmos inteligentes, gerando flexibilidade e autonomia em sua operação e que seja replicado em alta escala para as empresas? ”

4.2. ETAPA 2 – GERAÇÃO DE ALTERNATIVAS

Para a segunda etapa da pesquisa, a geração de alternativas, os pesquisadores se concentraram em como é possível ter uma evolução nos sistemas de planejamento de produção avançados que sejam aceitáveis no mercado e que resolvam os problemas identificados na primeira etapa. Para isso, esta etapa foi dividida em quatro sub-etapas. A primeira sub-etapa envolveu a realização da técnica de *Brainstorming* para identificar as funcionalidades e requisitos do sistema. A segunda sub-etapa envolvia a estruturação de um artigo sobre a aplicação prática da metodologia *DSR*. A terceira sub-etapa consistiu em uma pesquisa de mercado sobre as soluções atuais de APS, apontando pontos fortes e melhorias necessárias. Finalmente, a quarta sub-etapa apresentou um *mockup* do sistema, concentrando as informações levantadas previamente.

4.2.1. *Brainstorming*

A técnica de *Brainstorming* foi aplicada para coletar ideias e requisitos para um novo sistema de APS, baseado nas conclusões obtidas na primeira fase da pesquisa, que consistiu em uma Revisão Bibliográfica Sistemática (RBS). Algumas das funcionalidades identificadas incluem:

- a) Sistema com parâmetros e configurações simples e sem complexidade;

- b) Integração com sistemas de gerenciamento e execução da manufatura;
- c) Processamento por meio de algoritmos inteligentes, permitindo priorização conforme necessário;
- d) Interface intuitiva e clara, com flexibilidade na apresentação dos dados, facilitando a tomada de decisões;
- e) Visualização do sequenciamento e carga da máquina, com detalhamento ou agrupamento de acordo com as ordens de produção.

Estes requisitos foram levantados com o objetivo de melhorar a evolução dos sistemas de planejamento de produção avançados, garantir aceitação no mercado e resolver os problemas identificados na primeira fase da pesquisa.

O *brainstorming* também foi utilizado na etapa de *design*, no qual trata a concepção do produto.

4.2.2. DSR parcial sobre APS

A metodologia *DSR* foi aplicada parcialmente por Vieira, De Aguiar e Deschamps (2021) em um estudo sobre sistemas de planejamento de produção (*APS*). O artigo se concentrou na aplicação teórica da metodologia *DSR* no desenvolvimento de um sistema de *APS*, seguindo todas as etapas. No entanto, os pesquisadores sugeriram que o ciclo seja repetido em pesquisas futuras, abordando pontos não cobertos no estudo, como a integração com outros sistemas, o desenvolvimento do algoritmo de inteligência artificial e a análise de mercado, tanto na fase de definição do produto quanto na avaliação da usabilidade e do atendimento aos objetivos do sistema. O objetivo é desenvolver o sistema comercialmente e orientá-lo para a evolução dos sistemas *APS* no contexto da transformação digital e da Indústria 4.0.

4.2.3. Estudo de Mercado

Uma das atividades desta etapa 2 é a realização de um estudo de soluções de *APS* de mercado, buscando pontos fortes e pontos a desenvolver dessas soluções, com o objetivo de incrementá-los neste novo produto. O site G2 (<https://www.g2.com/>) apresenta 971 revisões de *softwares* de *APS*, geradas com base no *feedback* qualitativo e quantitativo de usuários reais. Atualmente, há 56 produtos de *APS* listados

no G2, e a TABELA 3 apresenta os mais populares, de acordo com as classificações e revisões dos usuários.

TABELA 3 – CLASSIFICAÇÃO SISTEMAS APS.

Produto	Classificação	Revisões
Acumatica	4,4	660
Soluções de fabricação SAP S/4HANA	4,2	53
Plataforma de Fabricação Inteligente da Plex	3,8	34
ERP de fabricação Katana	4,3	30
Plataforma de Fabricação Inteligente L2L	4,5	26

FONTE: Adaptado do G2 (2023).

A TABELA 4 apresenta as avaliações dos usuários sobre os produtos de APS, baseadas na experiência de usuário, configuração e suporte, utilizando uma escala numérica de 1 a 10.

TABELA 4 – CLASSIFICAÇÃO DOS USUÁRIOS SOBRE SOFTWARES APS.

Crítérios	Classificação do Usuário
Qualidade do Suporte	8,9
Facilidade de Uso	8,3
Facilidade de Configuração	8,1

FONTE: Adaptado do G2 (2023).

Além de avaliações com base em critérios dos usuários, os revisores também forneceram avaliações para as características e capacidades do *software*. As características mais bem avaliadas são apresentadas na TABELA 5.

TABELA 5 – AVALIAÇÃO DE CARACTERÍSTICAS DOS SOFTWARES DE APS.

Características	Classificação do Usuário
Inventário e Armazém	9,0
Distribuição	9,0
Manufatura	8,9

FONTE: Adaptado do G2 (2023).

Os usuários de também avaliaram as características mais específicas desses produtos. Na TABELA 6 estão apresentadas as características mais específicas que são consideradas mais importantes ao pesquisar as melhores ferramentas de *softwares* de APS.

TABELA 6 – CARACTERÍSTICAS MAIS IMPORTANTES EM SOFTWARES APS.

Características Mais Importantes	Classificação do Usuário
Chão de Fábrica	9,3
Ordens de Trabalho	9,2

FONTE: Adaptado do G2 (2023).

Os produtos de *softwares APS* no *website* G2 têm quase 217 tipos de recursos, todos os quais contribuem para o desempenho, usabilidade e funcionalidade do *software*. Essas capacidades de *software* são avaliadas e criticadas ao longo das 188 revisões de recursos criadas pelos usuários do G2. Essas pontuações e *feedback* detalhados podem ser úteis para determinar se um determinado produto de *software* de APS tem as características e funcionalidade ideais para cada seu negócio.

TABELA 7 – RECURSOS MAIS BEM AVALIADOS EM SOFTWARES APS.

Recurso Padrão	Classificação do Usuário	Características	Classificação do Usuário
Inventário e Armazém	90%	Produto Mestre	91%
		Previsão de Inventário	91%
		Custo de Inventário	91%
Distribuição	90%	Rastreamento	91%
		Expedição	90%
		<i>Picking & Packing</i>	88%
Fabricação	89%	Chão de Fábrica	93%
		Ordens de Trabalho	92%
		<i>Bill of Material</i>	91%

FONTE: Adaptado do G2 (2023).

Todas as ferramentas, aplicativos e produtos de APS possuem recursos padrão, e a TABELA 7 lista os mais bem avaliados, incluindo suas principais características que ajudam a impulsionar o desempenho desses recursos.

A TABELA 8 apresenta as funcionalidades mais importantes a serem consideradas na escolha de softwares e aplicativos de APS, segundo revisores do G2 e usuários do produto.

TABELA 8 – FUNCIONALIDADES MAIS IMPORTANTES DE SISTEMAS APS.

Funcionalidades Mais Importantes	Quantidade de Avaliações
<i>Picking & Packing</i>	34
Rotas	28

<i>Bill of Material</i>	18
Produto Master	18
Ordens de Trabalho	17

FONTE: Adaptado do G2 (2023).

Por fim, no QUADRO 3 são apresentadas as funcionalidades mais bem avaliadas e as funcionalidades que os usuários menos gostam em cada produto.

QUADRO 3 – FUNCIONALIDADES MAIS E MENOS AVALIADAS PELOS USUÁRIOS.

Produto	Funcionalidades com Maiores Avaliações	Funcionalidades com Menores Avaliações
Acumatica	Gestão da Qualidade	Funcionalidade
	Inventário & Armazém	Montagem
	Distribuição	Inventário
Katana Manufacturing ERP	<i>Back-Office</i>	Gestão de Inventário 3PL
	Fabricação	Contabilidade
	Gestão de Inventário	Gestão de Clientes
SAP S/4HANA	Agendamento	Planejamento

FONTE: Adaptado do G2 (2023).

4.2.4. Mockup do Sistema

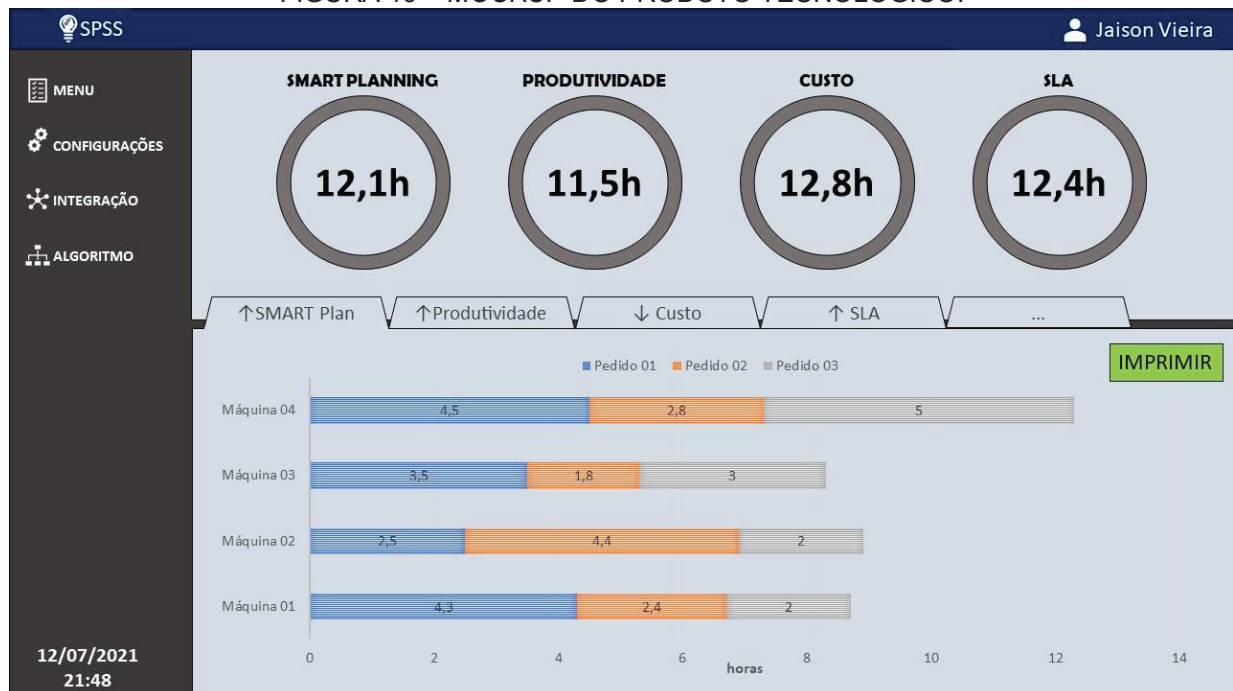
Para concluir a etapa 2, foi elaborado um primeiro protótipo do sistema com o objetivo de materializar todas as informações levantadas nas duas primeiras etapas. Este protótipo da solução foi elaborado no *PowerPoint* e pode ser visualizado na FIGURA 19. Esse protótipo servirá como base para o desenvolvimento posterior do sistema, permitindo que sejam feitas alterações e melhorias de forma mais eficiente.

Para conceber este protótipo, foram consideradas algumas informações cruciais que devem estar disponíveis para o usuário. A primeira é a visão da carga máquina em horas para cada tipo de objetivo priorizado na função *fitness*. Dessa forma, se o usuário quiser saber quantas horas de produção ele terá se priorizar, por exemplo, a programação com o objetivo de produtividade, essa informação estará disponível. Além disso, será exibida a carga máquina com a função de *Pareto* do algoritmo *NSGA-II*, que prioriza todos os objetivos de forma igual, gerando um cenário de programação inteligente.

A segunda visão é a visualização do gráfico de *Gantt* para cada cenário de programação, permitindo que o usuário expanda e visualize as ordens de produção ao longo do tempo, com seus tempos de fabricação, horários de início e fim e também em qual recurso produtivo deverão ser produzidas.

Por fim, o sistema deve apresentar um menu com a possibilidade de configurar a integração de dados com outros sistemas, como *ERP* e *MES*. Também será possível realizar configurações e cadastros com as informações da ordem de produção, estrutura de produto e roteiros de fabricação. Além disso, haverá um acesso para configurar o algoritmo que gerará os cenários de programação da produção.

FIGURA 19 – MOCKUP DO PRODUTO TECNOLÓGICO.



FONTE: O Autor (2023).

4.3. ETAPA 3 – DESENVOLVIMENTO DO ARTEFATO

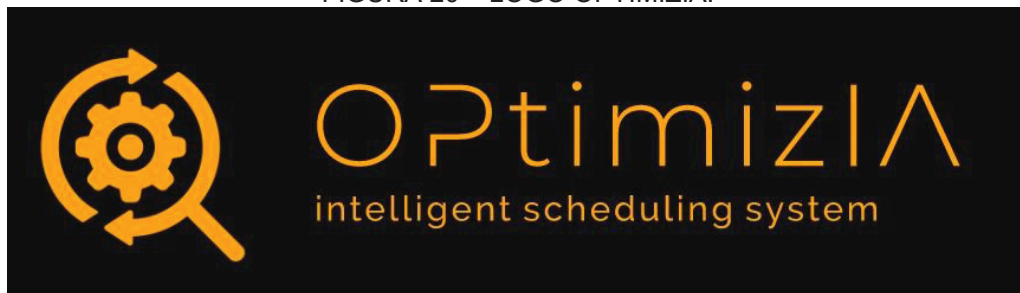
Em fase de desenvolvimento, o foco principal está no Produto Tecnológico, chamado de *OPTimizIA - intelligent scheduling system* (FIGURA 20). O nome do produto foi concebido a partir da palavra "otimização" e dos acrônimos "OP" (Ordem de Produção) e "IA" (Inteligência Artificial).

Como já mencionado na seção 1.5 deste trabalho, o projeto do sistema foi dividido em 4 etapas:

a) Definição da função objetivo;

- b) Coleta de dados e arquitetura do sistema;
 - c) Definição e estruturação do algoritmo inteligente;
 - d) Apresentação dos resultados da otimização (*design, front-end e back-end*).
- Em seguida, será apresentado o detalhamento de cada uma destas etapas.

FIGURA 20 – LOGO OPTIMIZIA.



FONTE: O Autor (2023).

4.3.1. Definição da função objetivo

A função objetivo é uma expressão matemática que representa o que se deseja otimizar em um problema de otimização. Ela é utilizada para avaliar a proximidade da solução encontrada com a solução ideal. Essa otimização é geralmente realizada com base em certas restrições, também conhecidas como problemas de produção, que são representadas por equações ou desigualdades que limitam os valores das variáveis. Essas restrições geralmente representam os limites impostos pelas condições operacionais e físicas que o problema deve seguir, conforme apresentado na QUADRO 4 de Corrêa, Nogueira e Caon (2019).

QUADRO 4 – RESTRIÇÕES PARA ANÁLISE DA CAPACIDADE FINITA.

Grupo	Restrições
Sistema Produtivo	Máquinas
	Mão de Obra
	Ferramentas
	Calendário
	Turnos de Trabalho
	Roteiros de Fabricação
	Velocidades de Operação
	Restrições Tecnológicas
	Tempos de <i>Setup</i>

	Matriz de Dependência
Demanda	Plano Mestre de Produção
	Carteira de Pedidos
	Previsão de Vendas
	Mudanças Quantidades
	Mudanças Prazos de Entrega
Situações Reais	Matéria-Prima Disponível
	Situação das Máquinas
	Manutenções Programadas
	Situação corrente de Ordens
	Filas Existentes
Parâmetros Tomada de Decisão	Prioridades a serem atendidas
	Objetivos a serem atingidos

FONTE: Adaptado de Corrêa, Nogueira e Caon (2019, p.345).

A função objetivo pode ser de vários tipos, como função de custo, função de lucro, função de desempenho, entre outras. Ela pode ser linear, não-linear, de minimização ou maximização. Em geral, a escolha da função objetivo é feita de forma a representar o objetivo geral do problema a ser resolvido.

O sistema proposto utiliza Algoritmos Genéticos para resolver problemas de otimização de funções objetivas. Esses algoritmos são técnicas de otimização baseadas nos princípios da evolução natural, que ajudam a encontrar soluções ideais para problemas complexos. Este sistema, especificamente, usa Algoritmos Genéticos para resolver funções objetivas relacionadas ao planejamento de produção, permitindo que gestores possam se concentrar tanto em objetivos internos, como custo de fabricação, produtividade e prazo de entrega, quanto em objetivos externos, como custo da moeda e preços de *commodities* (QUADRO 5).

QUADRO 5 – FUNÇÕES PARA ANÁLISE DO ALGORITMO DE INTELIGÊNCIA ARTIFICIAL.

Funções	Objetivo
Produtividade	Maximizar
Margem de Contribuição	Maximizar
Custos de Fabricação	Minimizar
Data de Entrega	Atender
Atrasos	Minimizar
Estoques de Matéria-Prima	Minimizar
Estoques de Produto Acabado	Minimizar

Data de Vencimento	Minimizar
Dólar Matéria Prima	Minimizar
Dólar Produto Acabado	Maximizar
<i>Makespan</i>	Minimizar
<i>TWET</i>	Minimizar

FONTE: O autor (2023).

Além disso, o sistema proposto deve fornecer visões específicas para o planejamento, permitindo que os gestores possam priorizar e otimizar objetivos relevantes para sua área. Durante a execução dos planos, o algoritmo deve ser capaz de priorizar essas funções, maximizando-as ou minimizando-as, além de oferecer uma visão inteligente do planejamento da produção e análise da capacidade finita. Dessa forma, o sistema se adapta e busca a solução ideal para as funções objetivas, levando em consideração as restrições e requisitos específicos.

Em resumo, a função objetivo é a expressão matemática que define o que deve ser otimizado. Ela é a base para encontrar uma solução ótima para um problema de otimização, considerando as restrições impostas. No sistema *OPTimizIA*, a função objetivo é tratada como a função de aptidão, também conhecida como função *fitness*. O objetivo é equilibrar dois critérios conflitantes, o tempo total de conclusão (*makespan*) e o atraso e antecedência ponderado total (*TWET*).

4.3.2. Coleta de Dados e Arquitetura do sistema

A integração, coleta e tratamento de dados é uma etapa crucial no desenvolvimento de qualquer projeto tecnológico. É através da integração de dados que é possível combinar dados de diferentes fontes em um único sistema ou banco de dados. Isso pode incluir dados de diferentes sistemas internos, bases de dados externas, arquivos de mídia e outras fontes. A integração de dados é importante para permitir a análise e o uso eficiente dos dados em uma variedade de aplicações e para a criação de modelos e algoritmos. Porém, é importante que esses dados sejam coletados de forma precisa, completa e representativa para garantir sua qualidade e confiabilidade.

Já a estruturação de dados é o processo de organizar e transformar os dados coletados de forma que eles possam ser compreendidos, armazenados e usados de

quem exibe esses dados e o resultado da simulação. A arquitetura do sistema está representada na FIGURA 21 e será detalhada a seguir.

4.3.2.1. Serviço de Integração de Dados

Um sistema *APS* necessita de dados oriundos de outros sistemas, do qual basicamente consistem em *input* (entrada) para o planejamento da produção.

A principal base de informações para o *APS* é o *ERP*, pois é ali que se concentram as informações sobre o cadastro dos itens, a estrutura destes itens, roteiros de fabricação, informações sobre estoques e sobre compras das matérias primas e informações sobre a produção, por outro lado o *MES* reporta em tempo real a execução das ordens de produção geradas no *APS* e traz informações importantes como os indicadores de produção, o ritmo de produção e disponibilidade de máquinas, que são importantes para o planejamento em tempo real e não ficar gerando planejamento apenas por dados teóricos.

Assim, o QUADRO 6 apresenta informações sobre os módulos principais de um sistema *ERP*. Esses módulos incluem as suas funcionalidades específicas, bem como os principais campos de dados que podem ser exportados para outro sistema, conhecido como *APS*.

QUADRO 6 – DADOS DO *ERP*.

Módulo	Funcionalidades	Tabelas
Engenharia	Contém os cadastros básicos de processo, produto, estrutura, operações, rede pert e roteiros.	Produto
		Estrutura
		Operação
		Rede Pert
		Roteiro
Produção	Contém os cadastros básicos e as informações relativas ao chão de fábrica para a produção	Turno
		Grade de Turno
		Calendário Feriados
		Área de Produção
		Grupo de Recursos
		Recursos
Vendas	Contém informações sobre as necessidades a serem planejadas.	Pedidos
		Previsão de Vendas

Estoque	Contém a informação sobre a quantidade disponível de cada item.	Estoque
		Depósitos
		Lote Econômico
		Lote Mínimo
Compras	Contém a informação de matéria-prima a serem recebidas.	Lead time de Entrega
		Lote de Compra
Custos	Contém a informação de custos dos itens.	Centro de Custo
		Custo unitário

FONTE: O Autor (2023).

E o QUADRO 7 apresenta informações sobre os módulos principais de um sistema *MES*. Esses módulos incluem as suas funcionalidades específicas, bem como os principais campos de dados que podem ser exportados para outro sistema, conhecido como *APS*.

No sistema *OPTimizIA* a integração de dados é realizada com o *ERP Protheus* e o serviço de integração é quem realiza a interface para gerenciar os dados que são necessários para o algoritmo.

QUADRO 7 – DADOS DO *MES*.

Módulo	Funcionalidades	Tabelas
Produção	Reporta quantidade de peças produzidas por ordem de produção em cada máquina, além do tempo real de ciclo por item.	Quantidades
		Tempos de Ciclo Real
Materiais	Reporta quantidades reais de matéria prima consumida na Ordem de Produção.	Consumo real
Manutenção	Contém os indicadores de máquinas para cálculo de carga máquina.	<i>MTTR</i>
		<i>MTBF</i>
		Disponibilidade

FONTE: O Autor (2023).

A integração com o *ERP Protheus* é realizada por um serviço de requisição de dados, o qual retorna a resposta com um arquivo *.CSV (Comma-Separated Value)* com os dados solicitados. O arquivo *.CSV* é uma forma comum de representar dados tabulares, com as colunas separadas por vírgulas. Esse arquivo é então utilizado pelo algoritmo para realizar as simulações e cálculos necessários.

4.3.2.2. Servidor do sistema

Um servidor é um dispositivo ou aplicativo responsável por fornecer recursos e serviços a outros dispositivos ou aplicativos conectados à rede. Esses recursos e serviços podem incluir: armazenamento de dados, processamento de dados, autenticação e autorização de usuários, gerenciamento de rede, além de outros.

Em um sistema, o servidor é responsável por receber e processar as solicitações de outros dispositivos conectados à rede, como computadores, dispositivos móveis e sensores. Ele pode armazenar e gerenciar grandes quantidades de dados, realizar cálculos complexos, fornecendo recursos e aplicativos aos dispositivos conectados.

A FIGURA 21 ilustra uma arquitetura de sistema que utiliza um servidor principal rodando o *framework back-end* chamado *Django*, que é baseado na linguagem de programação Python. Esse servidor principal contém um módulo chamado *Pandas*, que é utilizado para manipular e analisar os dados que são coletados do *ERP* através do arquivo *.CSV*. Em seguida gera um arquivo *JSON* que pode ser consumido pelo *front-end* e guarda os dados do arquivo na base de dados, que é o *PostgreSQL*.

4.3.2.3. Serviço de front-end

O *Front-end* é a parte do desenvolvimento de *software* que se concentra na interface do usuário, ou seja, a camada de apresentação de um sistema. Ele é responsável por criar e exibir a interface gráfica que os usuários interagem para acessar as funcionalidades do sistema. Ele é composto pelas tecnologias de marcação (*HTML*), estilo (*CSS*) e programação (*JavaScript*) e é responsável por criar a interação entre o usuário e o sistema.

O *front-end* é responsável por interpretar as informações que são enviadas pelo *back-end* e apresentá-las de forma amigável e fácil de usar para o usuário final.

O *front-end* também é construído utilizando o *framework Django*. Esse *front-end* utiliza três tecnologias para criar uma interface visual atrativa e interativa para o usuário final: Tailwind CSS, que é utilizado para estilizar a interface; HTML5, que é utilizado para estruturar as informações na página; e JavaScript, que é utilizado para

dar dinamicidade à interface, permitindo que as informações sejam atualizadas e manipuladas em tempo real.

4.3.3. Estruturação do algoritmo inteligente

Estruturação de um algoritmo inteligente é o processo de organizar as etapas do algoritmo de forma a garantir que o algoritmo possa ser executado corretamente e atingir o objetivo proposto. Isso inclui a definição das entradas, saídas, as regras e procedimentos do algoritmo, a forma como os dados serão processados e a lógica de tomada de decisão.

Estruturar um algoritmo inteligente também inclui a escolha de um método de aprendizado de máquina ou técnica de inteligência artificial apropriada para o problema a ser resolvido, seja supervisionado, não supervisionado, reforço, etc. Isso inclui também a identificação dos dados de treinamento e teste, a configuração dos parâmetros do algoritmo, a avaliação de desempenho e o ajuste do algoritmo de acordo com os resultados obtidos.

Por fim, estruturar um algoritmo inteligente também inclui a implementação do algoritmo em um sistema de software, testando e validando seu desempenho e fazendo ajustes conforme necessário.

Desta forma, o algoritmo selecionado para a solução é o Algoritmo Genético de Classificação Não Dominada II (*NSGA-II*), que é um algoritmo de otimização multiobjetivo que combina técnicas de algoritmos genéticos (AGs) e classificação não-dominante. Ele é projetado para encontrar uma solução Pareto-ótima para problemas multiobjetivos, ou seja, problemas com múltiplos objetivos que não podem ser melhorados simultaneamente. Segundo Kalyanmoy Deb (2002), o *NSGA-II* foi projetado como uma "solução rápida e elitista para otimização multiobjetivo".

O modelo do *NSGA-II*, desenvolvido por Wu (2018), foi selecionado para o servir como base para o projeto. A escolha foi feita pois o algoritmo possui operadores em suas etapas que já são adaptados especificamente para o problema de agendamento. O algoritmo retorna como resposta um conjunto de respostas representadas na fronteira primária do *NSGA-II*, que se referem ao sequenciamento de ordens de produção no tempo, otimizadas com o intuito de minimizar as funções objetivo *Makespan* e *TWET*. Entre os principais atrativos do algoritmo se destaca o seu

operador de mutação, onde é utilizada a mutação por deslocamento, que preserva a integridade dos alelos e altera apenas o sequenciamento das soluções, ideal para o problema. Também relevante, depois do cruzamento é utilizada uma função de reparo, que tem como objetivo restaurar eventuais ordens de produção em excesso ou faltantes para as soluções oriundas da combinação. Por fim, um dos elementos de maior destaque do algoritmo é a sua funcionalidade de armazenar o tempo gasto em cada ordem de produção e em cada máquina, que permite que a plotagem do calendário seja realizada respeitando as sequências de máquinas das ordens de produção, assim como possibilita o cálculo das funções objetivo para cada solução. Essa lógica e as adaptações necessárias para a implementação do algoritmo são detalhadas no desenvolvimento do trabalho.

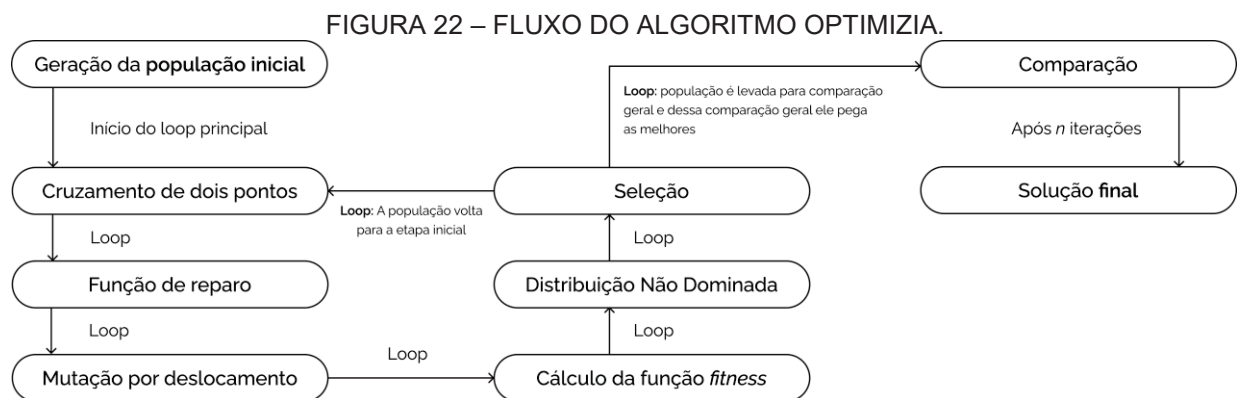
O *NSGA-II* funciona iterativamente, gerando novas soluções a cada iteração utilizando operadores genéticos como seleção, crossover e mutação. Em seguida, as soluções geradas são classificadas de acordo com sua dominação Pareto, ou seja, aquelas soluções que não são dominadas por outras soluções são consideradas como sendo de maior interesse. O algoritmo então seleciona as soluções não dominadas para a próxima iteração e atualiza a frente de Pareto. Este processo é repetido até que se alcance uma solução satisfatória.

Uma das principais vantagens do *NSGA-II* é que ele gera soluções diversas, o que é importante quando se busca soluções para problemas multiobjetivos, pois essas soluções podem ser usadas para decidir qual solução é mais adequada para o problema em questão. Segundo Mostaghim e Bader (2011) "a diversidade das soluções é uma característica desejável em problemas multiobjetivos, pois permite que o usuário escolha a solução mais adequada para o problema em questão".

De acordo com o artigo de Kalyanmoy Deb (2002), as etapas do algoritmo incluem:

- a) Geração inicial: é gerada uma população inicial de soluções candidatas ao problema.
- b) Classificação não-dominante: as soluções são classificadas de acordo com sua dominação Pareto, ou seja, aquelas soluções que não são dominadas por outras soluções são consideradas como sendo de maior interesse.
- c) Seleção: é realizada a seleção das soluções para a próxima iteração, geralmente utilizando algum método de seleção baseado em roleta ou torneio.

- d) Crossover e mutação: operadores genéticos são aplicados às soluções selecionadas para gerar novas soluções.
- e) Atualização da frente de Pareto: A frente de Pareto é atualizada com as novas soluções geradas e é usada para definir a população para a próxima iteração.
- f) Repetição: Essas etapas são repetidas até que se alcance uma solução satisfatória, ou seja, até que o critério de parada seja atingido.
- g) Resultado final: É gerado um conjunto de soluções não-dominadas, conhecido como frente de Pareto. Segundo o autor, "*NSGA-II* é projetado para gerar uma frente de Pareto de alta qualidade, o que é importante para problemas multiobjectivos, pois essas soluções podem ser usadas para decidir qual solução é mais adequada para o problema em questão".



FONTE: O Autor (2023).

O algoritmo do sistema proposto foi desenvolvido para otimizar a função objetivo. Ele é baseado no algoritmo genético *NSGA-II* e segue um fluxo estruturado, conforme ilustrado na FIGURA 22, e compreende as seguintes etapas:

1. Importação de dados: O algoritmo inicia com a importação de uma lista de ordens de produção, que é lida a partir de um arquivo .CSV. Este arquivo contém informações sobre as quantidades, operações e máquinas necessárias para cada ordem de produção.
2. Geração de soluções iniciais: Em seguida, o algoritmo gera uma série de soluções aleatórias para o agendamento das ordens de produção. A quantidade de soluções geradas é determinada pelo tamanho da população (*population_size*) e o número de agendamentos (*num_schedules*).

desejados. Estas soluções iniciais são representadas por listas de números, que indicam quando cada ordem de produção deve ser agendada.

3. Cruzamento: O algoritmo então realiza o cruzamento de dois pontos entre as soluções geradas aleatoriamente, a fim de gerar novas soluções, conhecidas como "filhas". Esse processo aumenta a diversidade da população e amplia o espaço de busca pela solução ótima. No entanto, as soluções geradas através do cruzamento podem apresentar agendamentos irregulares, o que requer uma segunda função de reparo.
4. Função de reparo: A primeira função de reparo é aplicada para garantir que cada ordem de produção apareça na lista de soluções com a frequência necessária. Isso é necessário, pois as soluções iniciais geradas aleatoriamente podem não conter o número de agendamentos adequado para cada ordem de produção.
5. Mutação: Para complementar o cruzamento, o algoritmo também realiza a mutação por deslocamento. Essa técnica introduz soluções que diferem totalmente da população inicial. A probabilidade de mutação (*mutation_rate*) e o percentual de agendamentos modificados na lista (*mutation_selection_rate*) são os inputs necessários para a realização da mutação.
6. Cálculo de função de aptidão: O algoritmo calcula a função de aptidão, também conhecida como função *fitness*, para avaliar a qualidade das soluções geradas. Essa função tem dois objetivos: minimizar o tempo de conclusão total (*makespan*), conforme equação (1), e o atraso e antecedência ponderado total (*TWET - Time Weighted Earliness and Tardiness*), conforme equação (2). Esses objetivos são conflitantes, pois o *Makespan* busca a conclusão mais rápida possível, enquanto o *TWET* busca a conclusão mais próxima da data de entrega. Portanto, é necessário fazer um *trade-off* entre essas duas soluções.

a. *TWET*:

$$\sum_n \left(\frac{1}{prioridade(OP)} \right) \times antecipação(OP) + (prioridade \times atraso(OP)) \quad (1)$$

b. *Makespan*:

$$\max (lista_tempos_totais_OP) \quad (2)$$

7. Atualização da frente de Pareto: O algoritmo atualiza a frente de Pareto, que é a coleção de soluções não-dominadas encontradas até o momento. Sempre na etapa de seleção são escolhidas prioritariamente as soluções das melhores fronteiras, no entanto no caso de apenas algumas soluções poderem ser selecionadas de uma fronteira (pelo número limitado do tamanho da população), o critério de desempate é dado para soluções de maior distância de aglomeração.
8. Seleção: Após o cálculo da função de aptidão, o algoritmo realiza a seleção das soluções. Esse processo é realizado utilizando a técnica de classificação não-dominante, que seleciona as soluções que não são dominadas por outras soluções. Essas soluções são consideradas de maior interesse e são selecionadas para a próxima iteração. Para todas as soluções de todas as fronteiras criadas é calculada a distância de aglomeração para o objetivo *makespan* e a distância de aglomeração para o objetivo *TWET*, que são somadas entre si para gerar a distância de aglomeração total de cada solução, usada para promover diversidade na busca por novas soluções.
9. Repetição do processo e Comparação: O processo é repetido até que se alcance uma solução satisfatória, ou até que se atinja o número máximo de iterações estabelecido previamente. Assim, com a seleção das melhores soluções, a partir da primeira iteração, o algoritmo salva estas melhores soluções em uma lista (*best_list*) e retorna para a etapa de cruzamento de dois pontos, sendo que a cada iteração do loop do algoritmo as melhores soluções da iteração atual são comparadas com as melhores soluções de todas as iterações anteriores, sendo realizada novamente a etapa de seleção com esse agrupamento. A evolução do resultado ocorre a cada loop que o algoritmo realiza, salvando sempre as melhores soluções, evoluindo o resultado final.
10. Saída: O algoritmo então gera uma saída com a solução Pareto-ótima encontrada, que é o conjunto de soluções que não podem ser melhoradas

simultaneamente em relação aos objetivos estabelecidos. No final, a melhor solução encontrada em todas as iterações será a saída. Como este é um problema multiobjectivo, pode haver vários conjuntos de soluções. A configuração definida aqui é ter o mesmo número de soluções que o tamanho da população.

O sistema proposto utiliza técnicas de algoritmo genético para otimizar o agendamento de ordens de produção, através da geração de soluções aleatórias, cruzamento, mutação, reparo e seleção, buscando equilibrar dois objetivos conflitantes, o tempo de conclusão total e o atraso e antecedência ponderado total.

4.3.4. Desenvolvimento da Solução (*design*, *front-end* e *back-end*)

O desenvolvimento do sistema foi realizado em três etapas. O *design* é a primeira etapa, onde são definidos os requisitos do sistema, a arquitetura geral, os fluxos de usuário e as interfaces. Isso inclui a criação de *wireframes*, protótipos e outros materiais de design para garantir que o sistema seja fácil de usar e atenda às necessidades do usuário final.

A segunda etapa é o desenvolvimento do *back-end*, onde são implementadas as funcionalidades do sistema, a integração com outros sistemas e a lógica de negócios. Isso inclui a escrita de código para garantir que o sistema seja escalável, confiável e seguro.

E por fim, a terceira etapa é o desenvolvimento do *front-end*, onde são implementadas as interfaces, isso inclui a criação de telas e a implementação de interações para garantir que o sistema seja fácil de usar e entender para o usuário final.

4.3.4.1. Desenvolvimento da Solução – Design

A etapa de *design* é uma das primeiras etapas no processo de desenvolvimento de um sistema. Ela é responsável por estabelecer as bases para o sistema, incluindo a definição dos requisitos, a concepção e a criação de protótipos (PRESSMAN, 2018).

Durante a etapa de *design*, os desenvolvedores se concentram em entender as necessidades dos usuários e em como o sistema deve funcionar para atender a

essas necessidades (SHNEIDERMAN, 2016). Isso inclui a realização de entrevistas e pesquisas com usuários potenciais (NIELSEN, 1994), bem como a análise de dados e estatísticas (SAUNDERS, 2009) para identificar quais são os problemas mais críticos que o sistema deve resolver.

FIGURA 23 – *BRIEFING* DO PROJETO PARA O TIME DE *DESIGN*.

Project Briefing



FONTE: O Autor (2023).

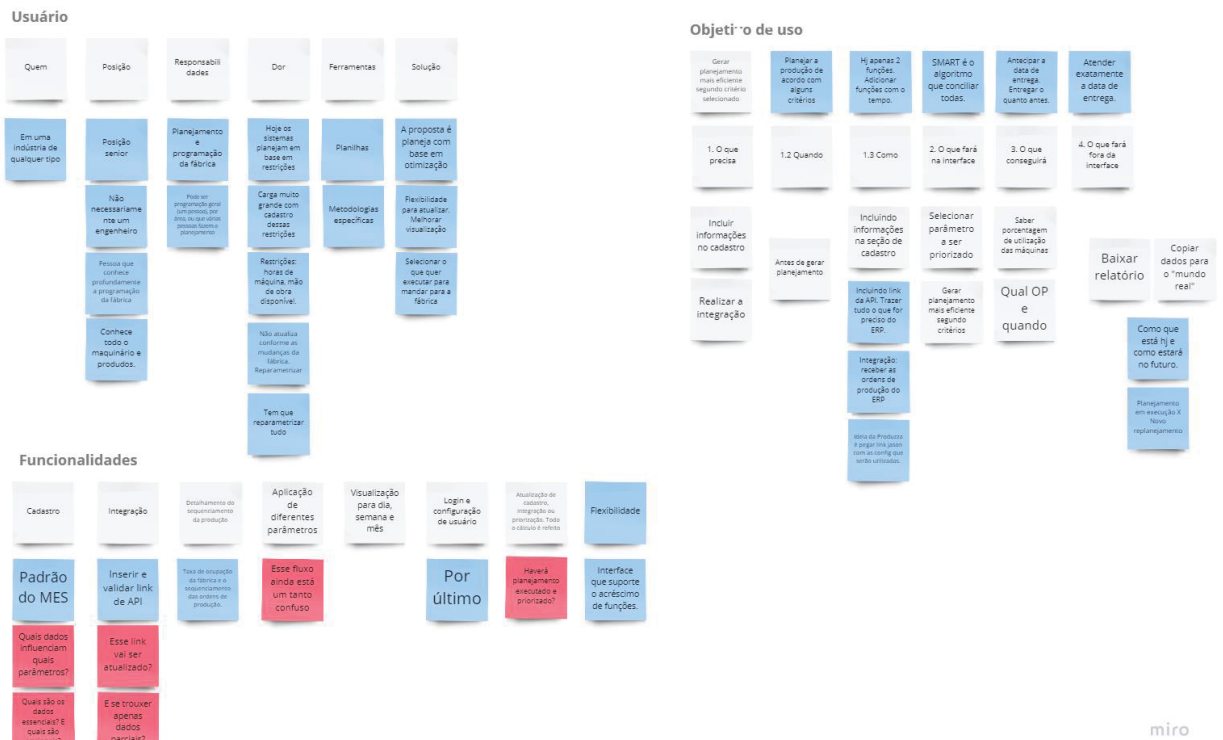
Para iniciar o desenvolvimento desta solução foi realizado um *briefing* sobre o projeto. Durante essa atividade foram feitas contextualizações sobre a aplicação da solução, assim como uma explicação do problema que seria resolvido, bem como, quem seriam os usuários do sistema e qual seria a solução desenvolvida. Uma imagem desta etapa pode ser visualizada na FIGURA 23.

Após o *briefing*, foi conduzida uma entrevista com o objetivo de obter uma compreensão mais profunda das características dos usuários, das funcionalidades desejadas da solução e dos objetivos de uso. As respostas obtidas ao longo da entrevista foram fundamentadas no estudo de revisão sistemática da literatura relevante e estão apresentadas na FIGURA 24.

Com o levantamento das informações iniciais, objetivos e funcionalidades desejadas, foi possível trabalhar com a proposta de valor da solução proposta. Isso incluiu o detalhamento do segmento de clientes, como suas atividades, dores e possíveis ganhos com a solução.

FIGURA 24 – ENTREVISTA COM USUÁRIOS.

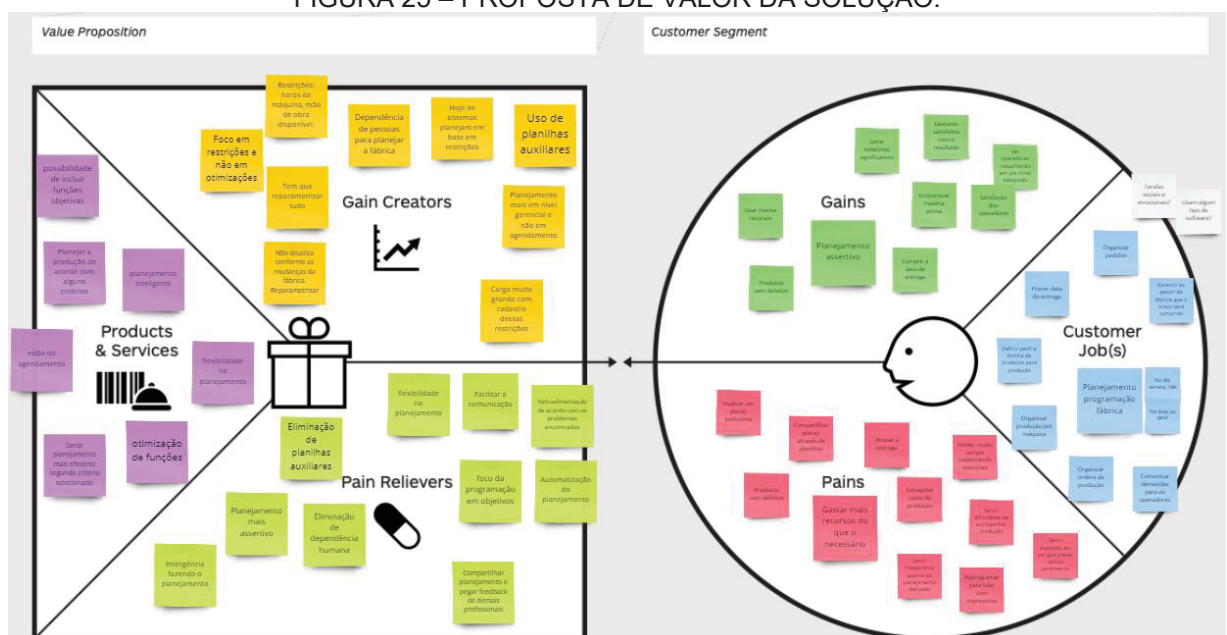
Stakeholder interview



FONTE: O Autor (2023).

Além disso, também foram detalhados o produto e serviço da solução, incluindo os criadores de ganhos e a abordagem para tratar as dores identificadas no segmento de clientes. Tudo isso está apresentado na FIGURA 25.

FIGURA 25 – PROPOSTA DE VALOR DA SOLUÇÃO.

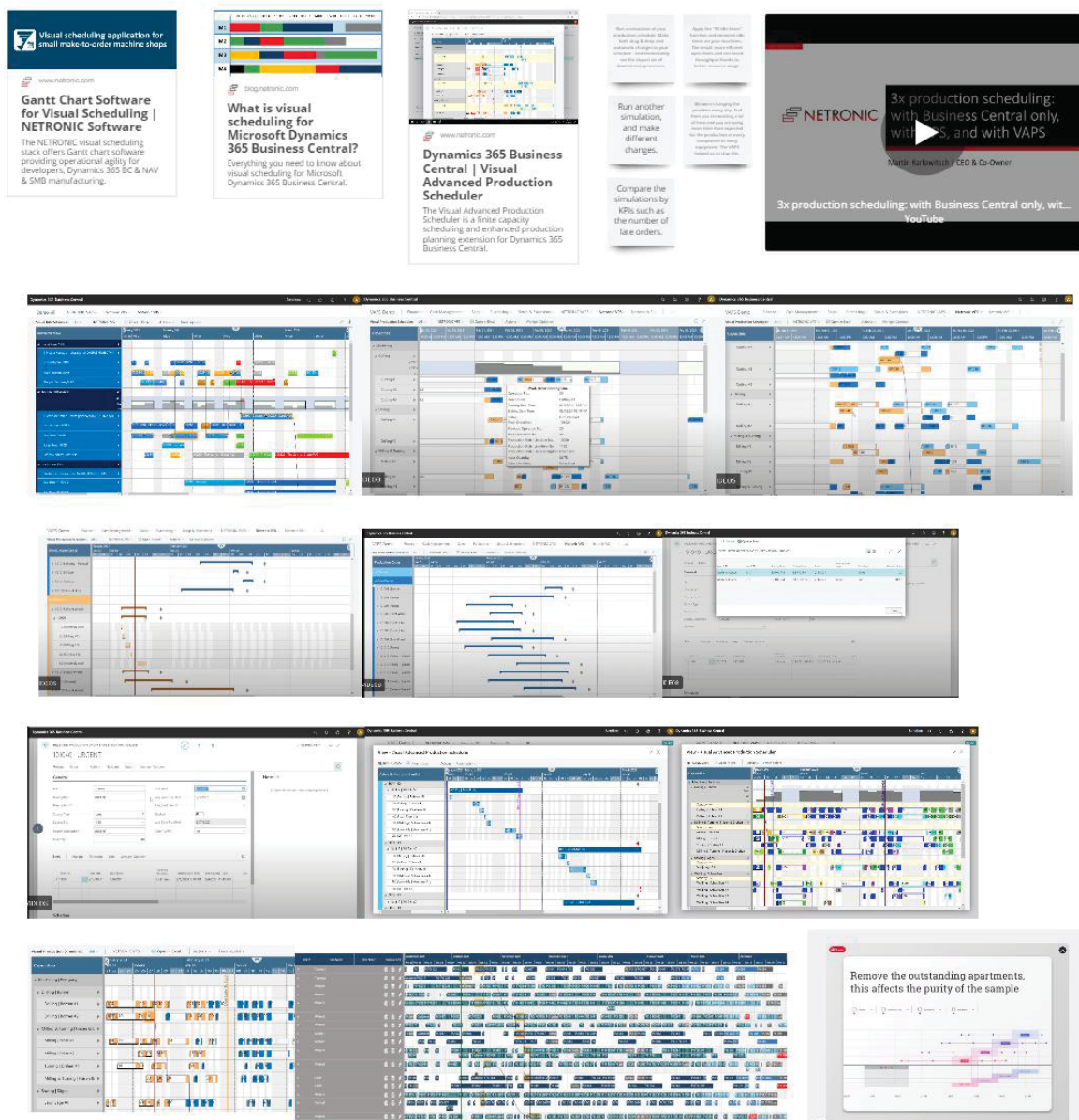


FONTE: O Autor (2023).

A próxima etapa do processo de design, conforme ilustrado na FIGURA 26 foi realizar uma pesquisa detalhada sobre as soluções concorrentes disponíveis no mercado. Durante essa pesquisa, foram analisadas as funcionalidades e características oferecidas pelos concorrentes, como a forma como as informações são apresentadas visualmente, a apresentação dos gráficos de agendamento e outros aspectos relevantes. Além disso, essa pesquisa também buscou identificar oportunidades de melhorias e ideias sobre como a solução poderia ser apresentada de maneira diferenciada e inovadora.

FIGURA 26 – COMPETIDORES DE SOLUÇÕES DE APS.

Competitors

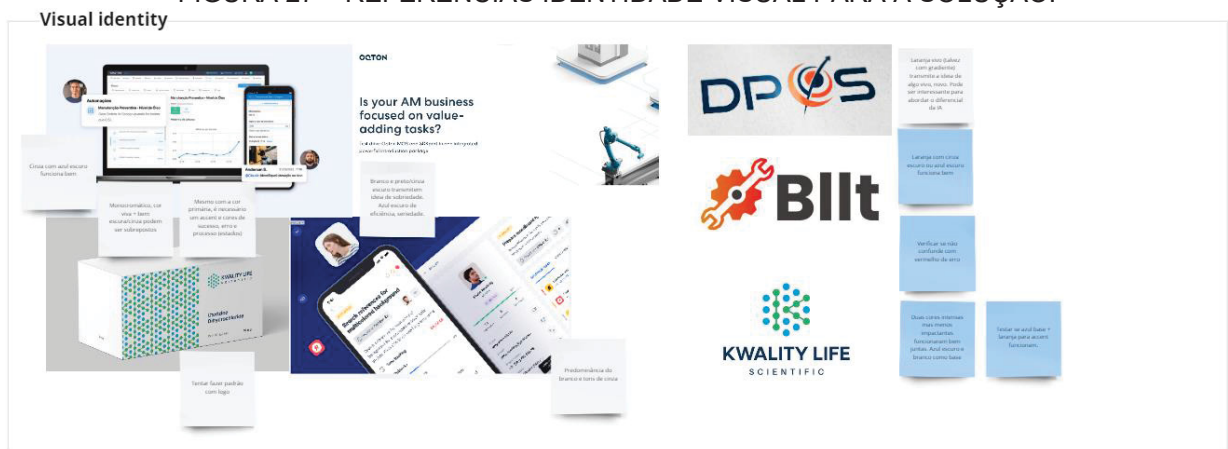


FONTE: O Autor (2023).

Com a pesquisa de soluções concorrente foi possível criar referências visuais. As referências visuais são uma parte importante no processo de design de um sistema, pois elas ajudam a garantir que a aparência e a funcionalidade do sistema sejam consistentes e fáceis de usar. Neste processo foram analisadas a identidade visual, os *dashboards*, a visão de agendamento e também o processo de *design*.

A Identidade visual (FIGURA 27) foi analisada através do conjunto de elementos visuais que compõem a aparência de um sistema, incluindo cores, fontes, ícones e layout.

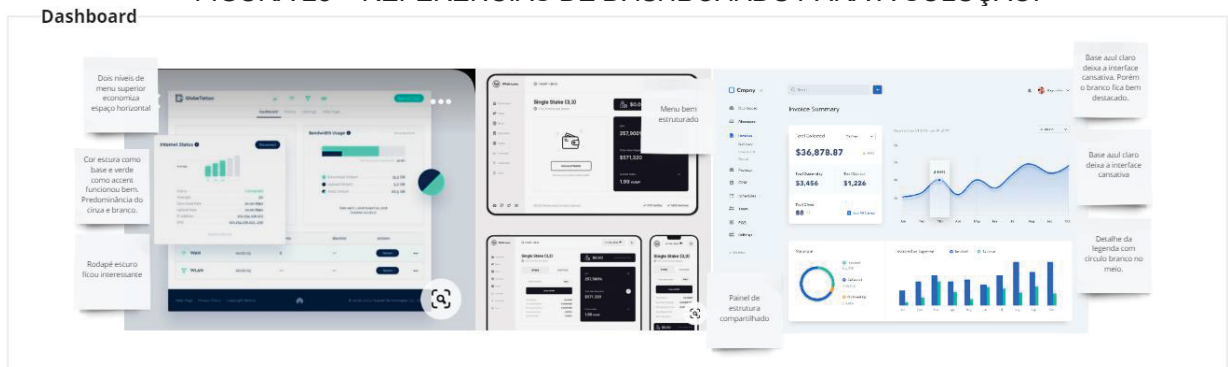
FIGURA 27 – REFERENCIAS IDENTIDADE VISUAL PARA A SOLUÇÃO.



FONTE: O Autor (2023).

Os *dashboards* (FIGURA 28) foram analisados com a visão geral das principais informações e métricas relevantes para o usuário.

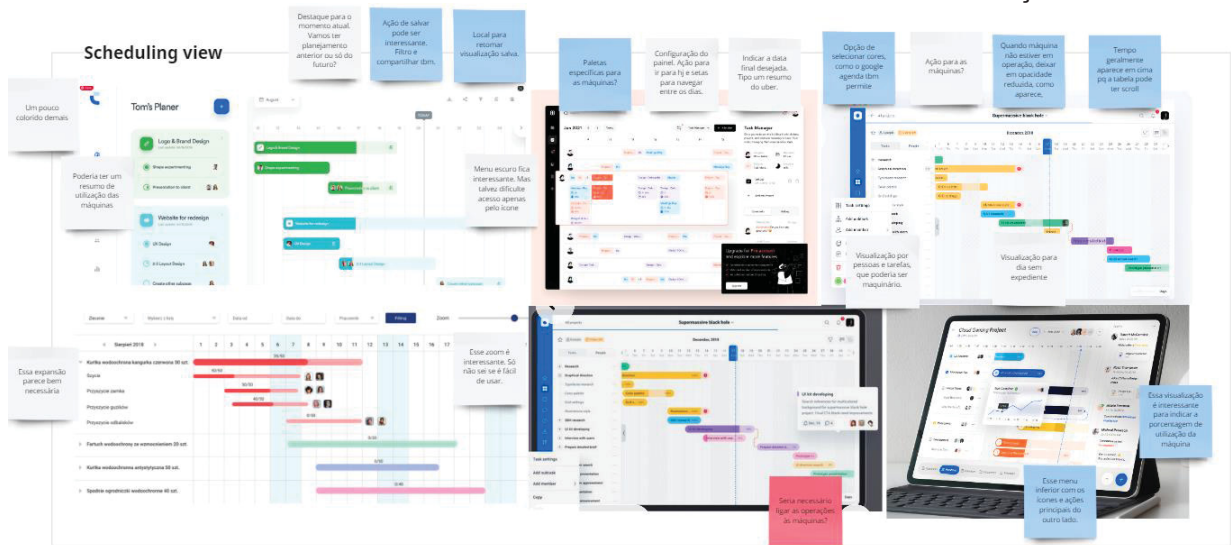
FIGURA 28 – REFERÊNCIAS DE DASHBOARDS PARA A SOLUÇÃO.



FONTE: O Autor (2023).

Já a *scheduling view* (FIGURA 29) foi analisada com a visão e gerenciamento das ordens de produção em um calendário ou tabela de horários e gráfico de *Gantt*.

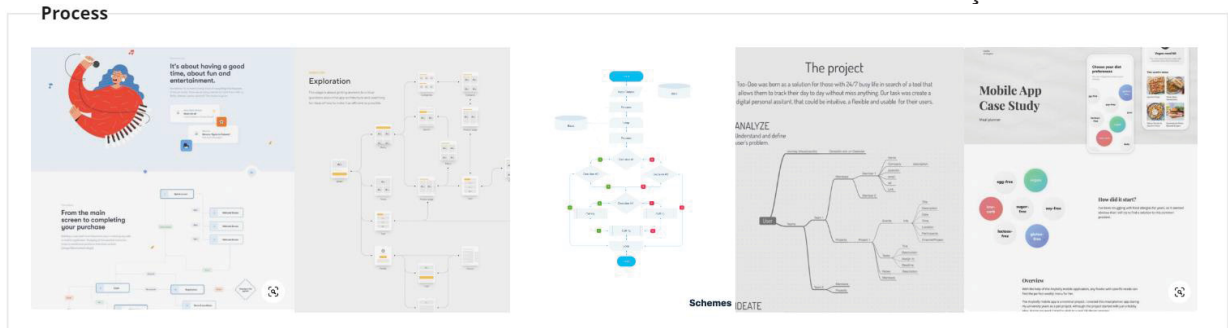
FIGURA 29 – REFERENCIAS DE GRÁFICO DE GANTT PARA A SOLUÇÃO.



FONTE: O Autor (2023).

E o processo de *design* (FIGURA 30) como deve ser realizado colaborativamente, envolvendo todas as partes interessadas, incluindo desenvolvedores, designers, gerentes de projetos e usuários.

FIGURA 30 – REFERENCIAS DE PROCESSOS PARA A SOLUÇÃO.



FONTE: O Autor (2023).

Com a conclusão do levantamento detalhado das informações e pesquisas sobre as soluções de mercado foi possível definir uma identidade visual (FIGURA 31) coerente e consistente para o sistema. A identidade visual é composta por um conjunto de elementos visuais, incluindo cores, fontes, ícones e layout, que contribuem para transmitir a personalidade e mensagem da marca, além de criar uma conexão emocional com os usuários, conforme destacado por Henderson (2014). Esses elementos foram cuidadosamente selecionados e aplicados para garantir que a identidade visual reflita a proposta de valor e os objetivos da solução proposta, e para que os usuários possam se relacionar de forma positiva e eficaz com o sistema.

FIGURA 31 – IDENTIDADE VISUAL.



A etapa de *design* também inclui a criação de protótipos de alto nível. Esses protótipos são versões simplificadas do sistema final, mas que servem para apresentar como o sistema funcionará e como os usuários interagirão com ele (NIELSEN, 1994; SHNEIDERMAN, 2016). O protótipo tem como objetivo apresentar uma visão geral do sistema e como ele será usado para auxiliar os usuários na tomada de decisões. Ele é baseado nos requisitos previamente levantados, e é desenhado para fornecer uma representação fiel do sistema final (PRESSMAN, 2018).

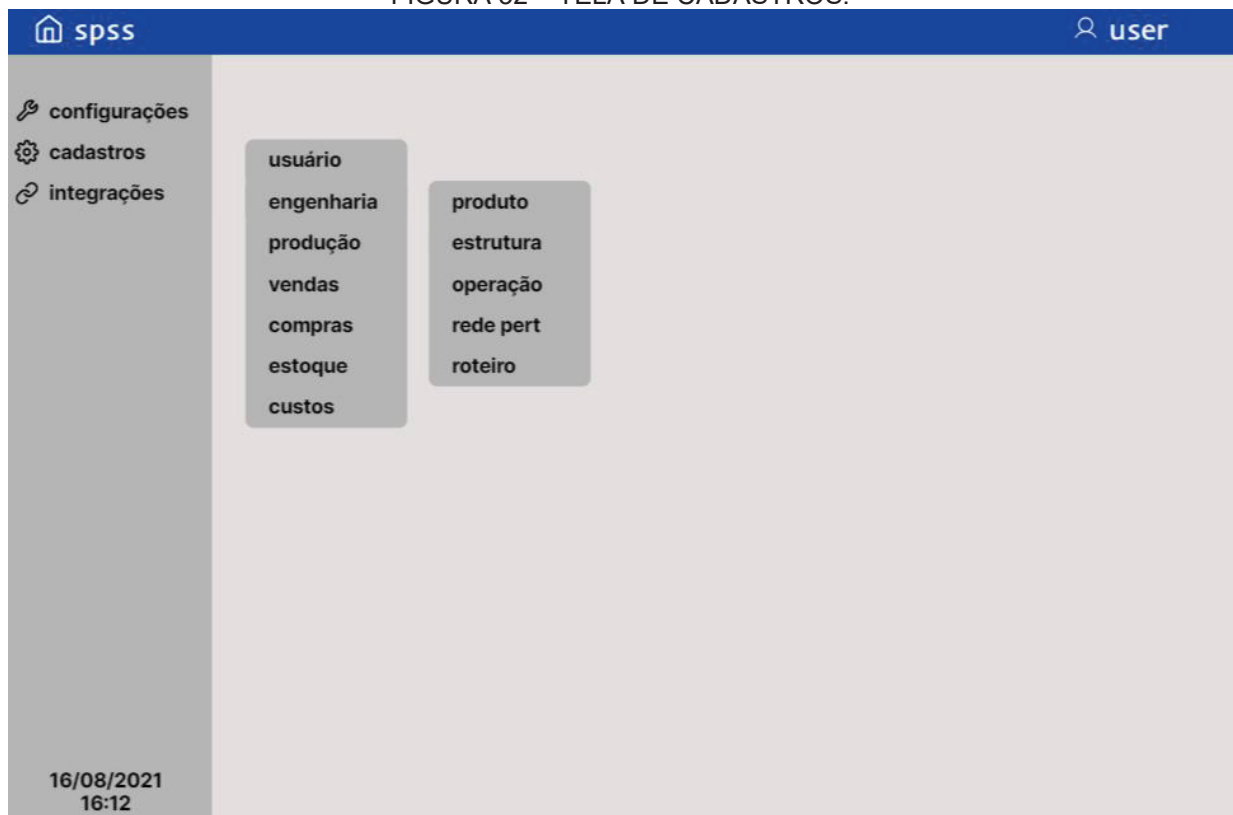
O protótipo inclui visualizações das informações de saída, como gráficos, tabelas e outros elementos que mostram como os dados serão apresentados aos usuários (KARAT, 1990). Além disso, ele também mostra como os usuários interagirão com o sistema, incluindo como eles acessarão e navegarão pelas informações. Seu desenvolvimento é uma ferramenta valiosa para garantir que o sistema final atenda às necessidades dos usuários e seja fácil de usar. Ele também permite que os desenvolvedores obtenham *feedback* dos usuários e façam ajustes antes de começar a desenvolver o sistema final, garantindo que o projeto esteja alinhado com as necessidades e expectativas dos usuários (HOLTZBLATT, 1998; COOPER, 1999). O uso de protótipos de alto nível também é uma técnica comumente utilizada no processo de design de interação humano-computador (HCI), pois permite que os designers

testem e validem suas ideias antes de implementá-las no sistema final (BUXTON, 2007).

Durante o processo de *design*, foram desenvolvidas três versões de protótipos. A primeira já apresentada na FIGURA 19, corresponde a ideia conceitual e de como seria a apresentação do algoritmo visualmente para os usuários e foi desenvolvido no *software Power Point*.

Já a segunda versão foi desenvolvida na plataforma *Framer* e já apresenta uma ideia mais avançada do sistema, com possibilidade de interação e apresentando tela de cadastros (FIGURA 32), tela de configuração (FIGURA 33) da integração, tela da parametrização do algoritmo (FIGURA 34), a tela com a visão de planejamento inteligente apresentada pelo algoritmo (FIGURA 35) e uma tela com uma visão de *dashboard* (FIGURA 36). Estas telas conceituais foram utilizadas como base pelo *designer* profissional para desenvolver os protótipos finais com alto nível de fidelidade ao sistema.

FIGURA 32 – TELA DE CADASTROS.



FONTE: O Autor (2023).

FIGURA 33 – TELA DE CONFIGURAÇÃO DA INTEGRAÇÃO.

spss user

configurações
cadastros
integrações

url erp:

☒ engenharia ☒ estoque
☒ produção ☒ Custos
☒ vendas
☒ compras

url spss:

☒ engenharia ☒ estoque
☒ produção ☒ custos
☒ vendas
☒ compras

url mes:

☒ produção
☒ indicadores

16/08/2021
16:12

FONTE: O Autor (2023).

FIGURA 34 – TELA DE PARAMETRIZAÇÃO DO ALGORITMO.

spss user

configurações
cadastros
integrações

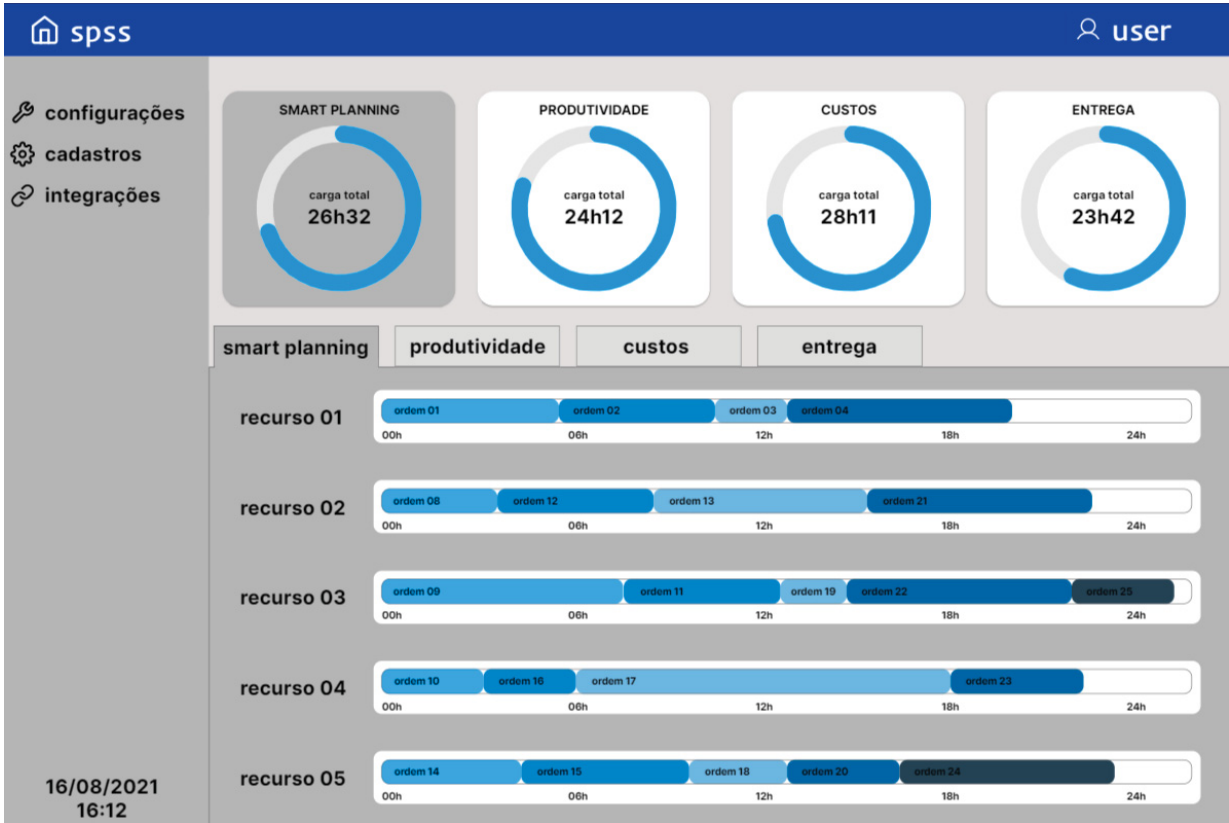
priorização do algoritmo:

1. maior produtividade
2. menor custo
3. data de entrega
4. maior margem
5. data de vencimento

16/08/2021
16:12

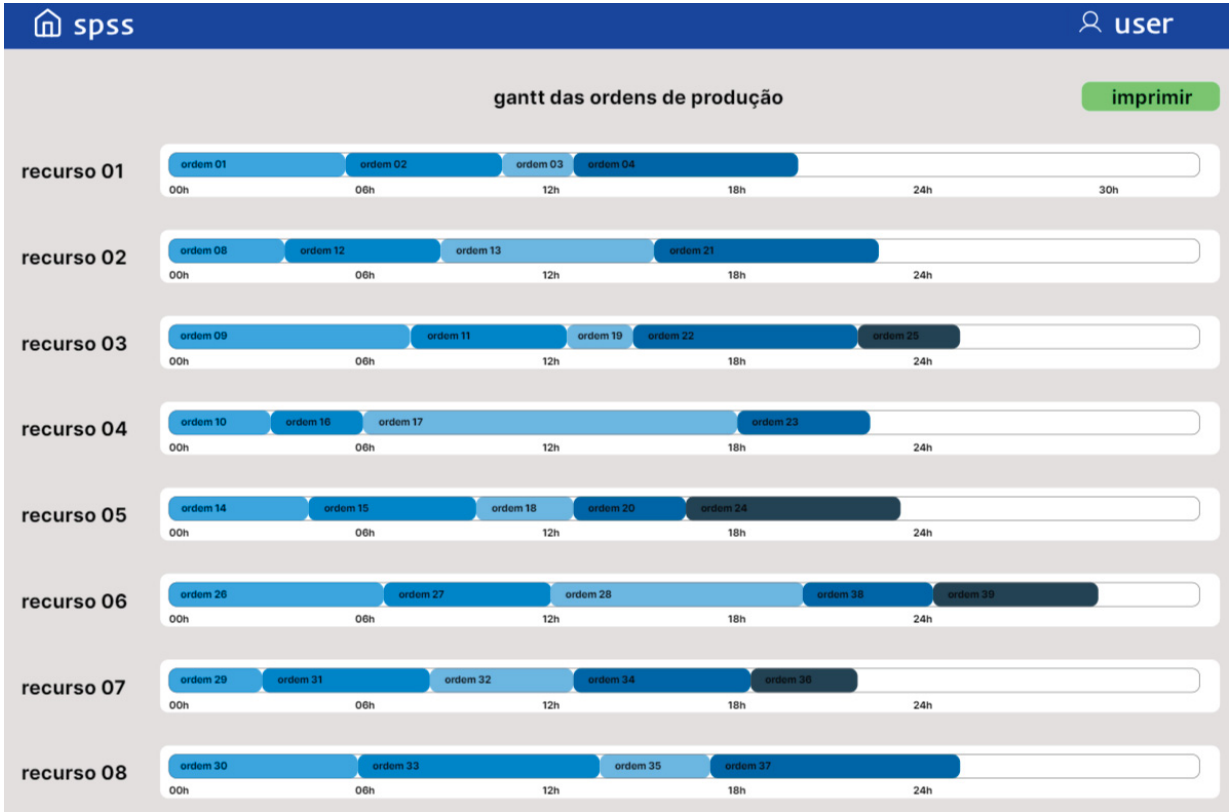
FONTE: O Autor (2023).

FIGURA 35 – TELA DE VISÃO DO PLANEJAMENTO INTELIGENTE.



FONTE: O Autor (2023).

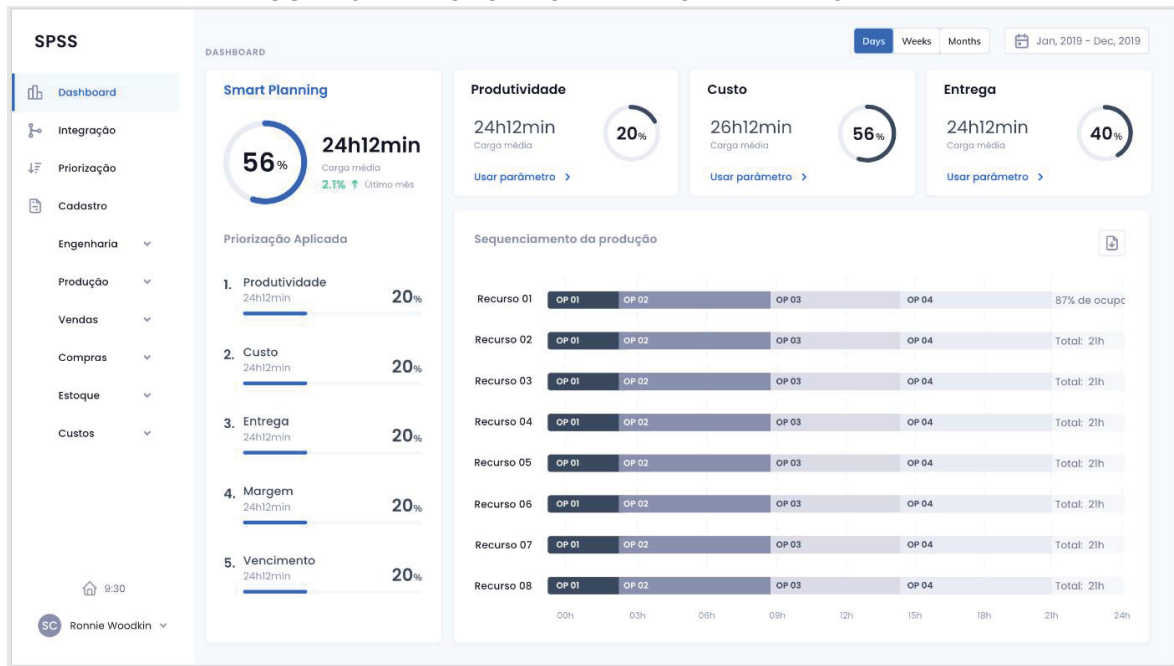
FIGURA 36 – TELA DE DASHBOARD DO PLANEJAMENTO.



FONTE: O Autor (2023).

Já a versão final do protótipo foi desenvolvida no *software Adobe XD*, por *designer* industrial profissional, este protótipo de alto nível de fidelidade é o protótipo utilizado para o desenvolvimento do *front-end* e está apresentado na FIGURA 37.

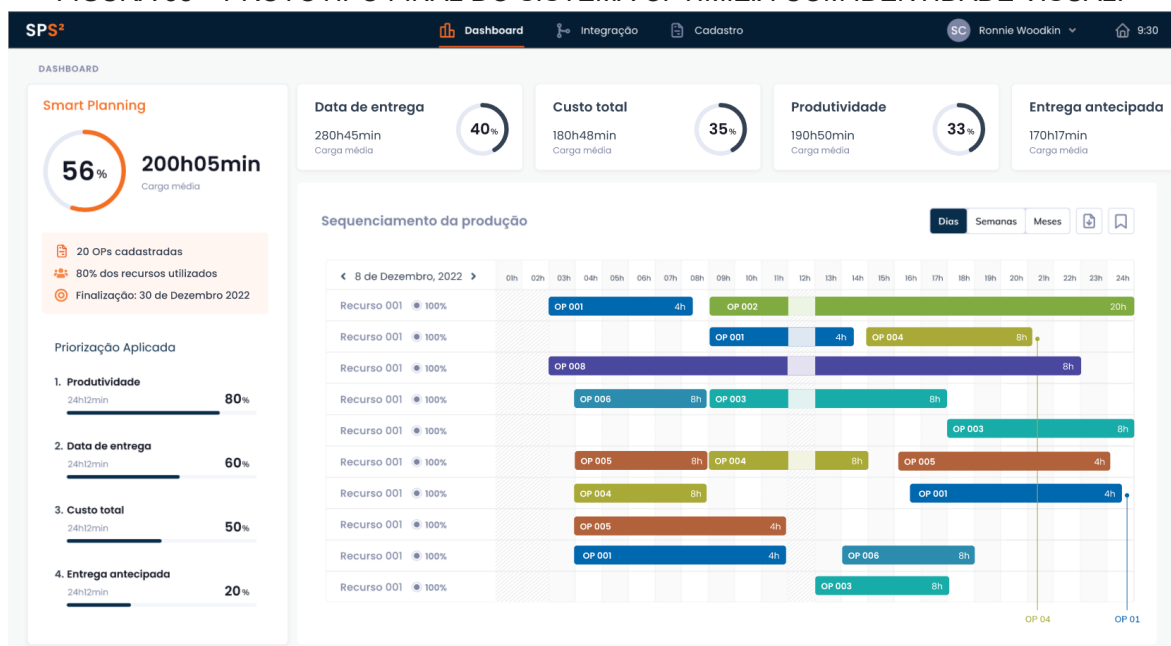
FIGURA 37 – PROTÓTIPO FINAL DO SISTEMA *OPTIMIZIA*.



FONTE: O Autor (2023).

Além disso há mais uma versão do protótipo, desta vez, com o propósito da aplicação da identidade visual do sistema e pode ser vista na FIGURA 38.

FIGURA 38 – PROTÓTIPO FINAL DO SISTEMA *OPTIMIZIA* COM IDENTIDADE VISUAL.



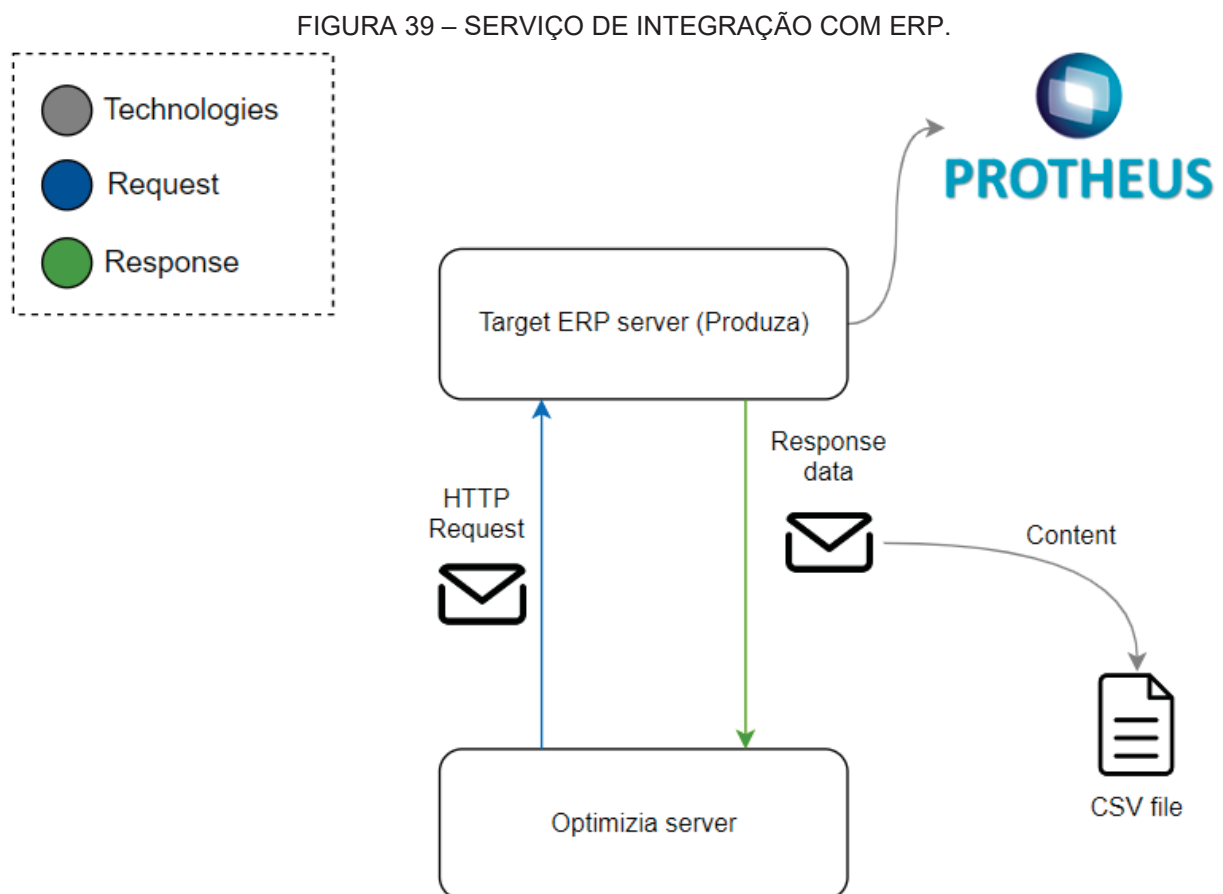
FONTE: O Autor (2023).

4.3.4.2. Desenvolvimento da Solução - Back-end

Em um sistema, o servidor é responsável por receber e processar as solicitações de outros dispositivos conectados à rede, como computadores, dispositivos móveis e sensores. Ele pode armazenar e gerenciar grandes quantidades de dados, realizar cálculos complexos, fornecendo recursos e aplicativos aos dispositivos conectados.

4.3.4.2.1. Back-end – Tecnologias e Arquitetura.

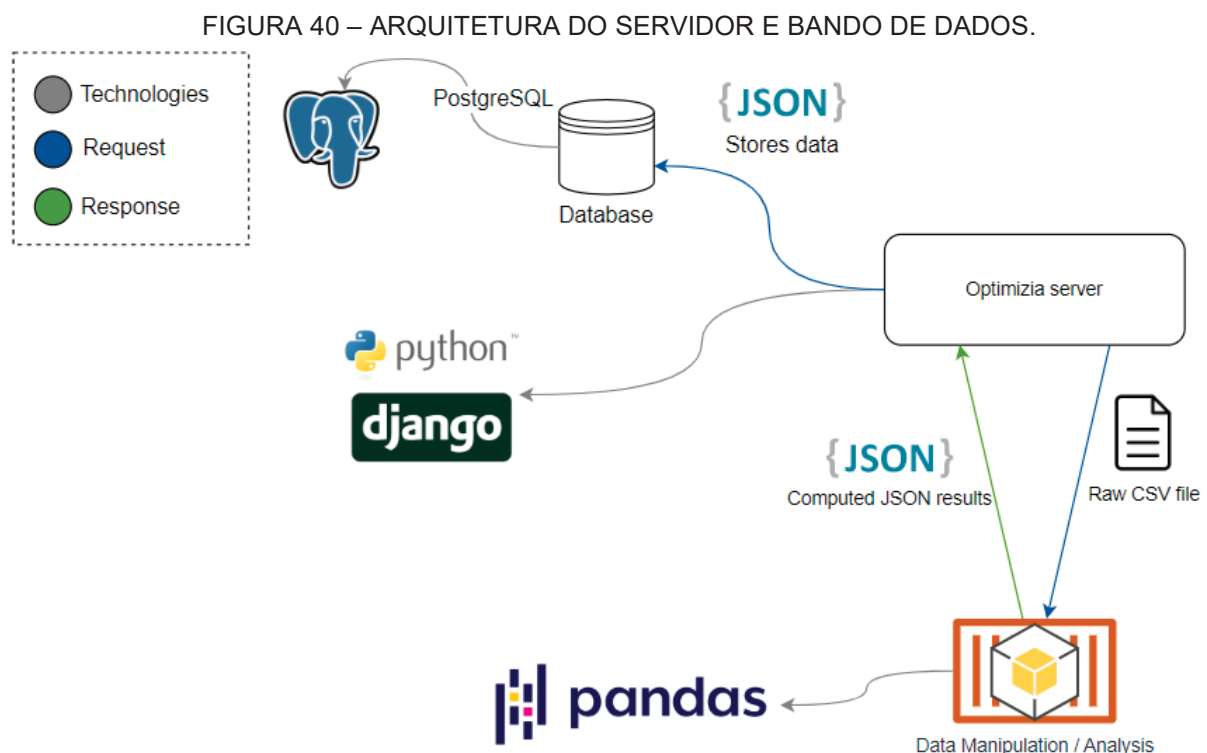
Para que o algoritmo inicie a otimização, é necessário carregar os dados no sistema, que consistem basicamente nas ordens de produção ativas. Normalmente, essas ordens são geradas dentro de um sistema *ERP*. No estudo de caso em questão, o sistema *ERP* é o *Protheus* e a empresa é a *Produza*. Assim, é necessário ter um serviço que realize a integração de dados entre o *ERP* e o sistema *OPTimizIA*.



FONTE: O Autor (2023).

A FIGURA 39 representa a arquitetura de integração com o sistema ERP. Assim, a primeira etapa é o serviço de requisição de dados para o através de um *HTTP request* (*Hypertext Transfer Protocol*). Esse serviço tem como objetivo enviar uma solicitação para o ERP Protheus, utilizando o protocolo *HTTP*, para obter os dados necessários para o algoritmo. Em seguida o serviço realiza a resposta da requisição, onde o *ERP* envia um arquivo *.CSV* (*Comma-Separated Value*) com os dados solicitados. O arquivo *.CSV* é uma forma comum de representar dados tabulares, com as colunas separadas por vírgulas. Esse arquivo é então utilizado pelo algoritmo para realizar as simulações e cálculos necessários.

A FIGURA 40 representa a arquitetura do servidor do sistema. Nesse contexto, o servidor principal recebe o arquivo *.CSV* e o encaminha para o módulo *Pandas*, o qual realiza a manipulação e análise dos dados. O *Pandas* é uma biblioteca amplamente utilizada em *Python* para processamento de dados, sendo especialmente útil para lidar com grandes quantidades de dados tabulares.



FONTE: O Autor (2023).

O resultado dessa análise é então enviado ao servidor principal, que utiliza o *framework Django*, onde é realizado a escrita do código, conforme FIGURA 41, em seguida disponibiliza as informações em um arquivo *JSON* (*JavaScript Object*

Notation). Esse formato é adequado para ser facilmente consumido pelo *front-end* do sistema. Por fim, os dados do arquivo *JSON* são armazenados na base de dados *PostgreSQL*, a qual é utilizada para garantir a integridade e disponibilidade dos dados gerados pelo sistema.

FIGURA 41 – ESCRITA DO ALGORITMO EM BACK-END.

```

1 import pandas as pd
2 import numpy as np
3 import numpy
4 from datetime import datetime
5 import time
6 import copy
7
8 ''' ===== initialization setting ===== '''
9
10 # Adaptações de dados
11
12 data = pd.read_excel('case_produza_rev01.xlsx', skiprows=1, usecols="B:H,J:K")
13 data.drop(data[data['OF'].isnull()].index, inplace=True)
14 data['OF'] = data['OF'].astype(int)
15 data['OF'] = data['OF'].apply(lambda x: 1 if x == 'RETRABALHO' else x) ----> caso deseje alterar 'RETRABALHO' pra outro nome
16 num_job=data['OF'].nunique() # number of jobs
17 num_mc=data['Processo'].nunique() # number of machines
18
19 #----- Tratativa para a base de dados SEQUENCIAMENTO de MÁQUINAS ms_tmp -----
20 ms_tmp=data[['OF','Processo']].copy()
21 for i in data.index:
22
23     245, 32, 172, 35, 115, 93, 48, 183, 121, 124, 165, 145, 118, 144, 119, 41, 109, 38, 70, 53, 159, 14, 66, 50, 33, 18, 151, 136, 234, 96, 216, 114, 82,
24     179, 39, 161, 206, 150, 123, 56, 72, 131, 128, 134, 167, 244, 168, 126, 62, 190, 129, 97, 174, 153, 117, 200, 171, 26, 20, 111, 130, 176, 58, 163, 156,
25     194, 232, 63, 112, 44, 162, 188, 233, 147, 157, 99, 141, 90, 158, 24, 213, 212, 195, 160, 77, 27, 192, 17, 106, 3, 2, 241, 31, 217, 76, 193, 22, 191, 22
26     0, 235, 218, 181, 240, 170, 226, 202, 42, 286, 51, 143, 15, 101, 54, 187, 137, 169, 209, 94, 205, 214, 142, 46, 133, 196, 12, 175, 23, 7, 198, 50, 182,
27     89, 98, 166, 203, 207, 173, 61, 204, 92, 210, 75, 231, 105, 67, 36, 55, 208, 9, 189, 1, 113, 223, 105, 140, 19, 16, 107, 152, 64, 210, 148, 104, 211, 22
28     7, 237, 215, 182, 84, 225, 88, 67, 242, 183, 83, 164, 40, 4, 230, 155, 29, 68, 199, 57, 243, 73, 229, 21, 221, 229, 11, 135, 184, 224, 149, 146, 222, 69
29     , 180, 177, 10, 30, 13, 6, 154, 236, 79, 5, 46, 47, 94, 46]]
30 [[1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 237284
31 4], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 237
32 2844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 237
33 2844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 2372844], [1697262070.1249342, 237
34 2844]]
35 the elapsed time:88.08361148834229
36 PS C:\Users\jve\Documents\Github\project_APS_IA>
  
```

FONTE: O Autor (2023).

4.3.4.2.2. Detalhamento do Algoritmo do Sistema

FIGURA 42 - FORMATO DA BASE DE DADOS FORNECIDA.

Prioridades	Roteiro	Processo	Sequência	Data	Tempo previsto em horas	OF	Código Placa	Descrição	Qtd	Tempos Unitários
1	1	SMD	1	23/01/23	2:36:27	00720601001	5000.001624	PCB CARTAO DTCV-01C.P.4	149	00:01:03
4	4	SMD	1	25/01/23	7:52:30	00747901001	5000.001711	MOD PCI CPU REV.05 HORUS V3.28 4G PLS62-W DB9F	450	00:01:03
4	4	VERNIZ	2	25/01/23	7:15:00	00747901001	5000.001711	MOD PCI CPU REV.05 HORUS V3.28 4G PLS62-W DB9F	450	00:00:58
10	1	SMD	1	31/01/23	1:45:00	00737201001	5000.001686	PCB 00014_04 INTERFACE GER_SI R08 024579	100	00:01:03
11	1	SMD	1	31/01/23	0:26:15	00737003001	5000.001533	PCB 00033_02 DRIVER MOTOR MAMMO R06 010621	50	00:00:32
12	3	SMD	1	31/01/23	2:35:00	00731802001	5000.001332	PCB 00055_03 CPU CONTROLE ARCO R01 017420	50	00:03:06
12	3	PTH	2	31/01/23	2:04:00	00731802001	5000.001332	PCB 00055_03 CPU CONTROLE ARCO R01 017420	50	00:02:29
12	3	INSPEÇÃO	3	31/01/23	2:19:30	00731802001	5000.001332	PCB 00055_03 CPU CONTROLE ARCO R01 017420	50	00:02:47
13	1	SMD	1	31/01/23	2:35:00	00731605001	5000.001329	PCB 00050_01 FONTE FILAMENTO ARCO R01 012813	50	00:03:06
14	1	SMD	1	31/01/23	0:26:15	00731808001	5000.001535	PCB 00007_03_VB FONTE FILAMENTO R04 010758	50	00:00:32
15	2	PTH	1	31/01/23	2:35:00	00731604001	5000.001333	PCB 00057_02 POTENCIA ARCO P02 016729	50	00:03:06
16	1	SMD	1	31/01/23	2:35:00	00731602001	5000.001334	PCB 00058_02 ALIMENTACAO TROLLEY ARCO P02 016725	50	00:03:06
17	1	SMD	1	31/01/23	1:15:00	00731603001	5000.001328	PCB 00049_02 ENTRADA ALIMENTACAO ARCO P02 016911	50	00:01:30
18	4	SMD	1	31/01/23	2:30:00	00731804001	5000.001567	PCB 00052_03 LEITURA KV ARCO P03 024685	50	00:03:00
18	4	VERNIZ	2	31/01/23	2:15:00	00731804001	5000.001567	PCB 00052_03 LEITURA KV ARCO P03 024685	50	00:02:42
19	1	SMD	1	31/01/23	1:15:00	00731809001	5000.001327	PCB 00048_02 DRIVER IGBT ARCO R01 016910	50	00:01:30
20	1	SMD	1	31/01/23	0:37:30	00731807001	5000.001483	PCB 00031_02 INTERFACE SAIDA MAMMO R05 010765	50	00:00:45
21	1	SMD	1	31/01/23	1:15:00	00731601001	5000.001326	PCB 00045_02 CAPACITOR GIGA ARCO P02 016932	50	00:01:30
23	1	SMD	1	31/01/23	39:49:30	00734903001	5000.000980	PCB CARTAO DSLC-02H.C.12	1593	00:01:30
24	1	SMD	1	31/01/23	8:45:00	00714218001	5000.000802	PCB CARTAO TCAN01 WT R002 CA01240202	350	00:01:30
25	1	SMD	1	31/01/23	23:15:00	00714219001	5000.001384	PCB CARTAO TCAN01 NT R002 CA01240102	450	00:03:06

FONTE: Adaptado de empresa Produza (2023).

Para adaptar o algoritmo e implementá-lo na base de dados da empresa *Produza*, foi necessário realizar um tratamento inicial dos dados, convertendo-os para o formato padrão requerido pelo algoritmo. Essa conversão foi realizada internamente, utilizando a biblioteca *Pandas* do *Python*.

Em seguida, foi preciso ajustar as funções de geração da população inicial e de reparo, a fim de possibilitar o funcionamento do algoritmo para ordens de produção com diferentes quantidades de máquinas em seu roteiro. A base de dados fornecida pela empresa *Produza* continha 251 linhas de informações, como mostrado na FIGURA 42.

Na planilha, cada linha representa um agendamento necessário para uma etapa de fabricação de uma ordem de produção, identificada pela coluna "OF". A coluna "Processo" indica a máquina responsável pela etapa e o "Tempo previsto em horas" indica o tempo de execução da operação no formato HH:MM:SS. A coluna "Data" indica a data de entrega da ordem de produção no formato DD/MM/AA, enquanto a coluna "Roteiro" especifica a sequência de máquinas necessárias para a produção da ordem de acordo com os roteiros disponíveis, que estão listados na TABELA 9.

TABELA 9 - ROTEIROS POSSÍVEIS DE FABRICAÇÃO.

Roteiro	Processo	Sequência
1	SMD	1
2	PTH	1
3	SMD	1
3	PTH	2
3	INSPEÇÃO	3
4	SMD	1
4	VERNIZ	2
5	SMD	1
5	INSPEÇÃO	2

FONTE: Adaptado da empresa Produza (2023).

Com base nos dados e conhecimentos dos roteiros, iniciou-se o processo de tratamento dos dados. Vale ressaltar que não apenas a tratativa foi realizada, mas também a atribuição automática de informações importantes da planilha para algumas variáveis, como o número de máquinas (*num_mc*) e o número de ordens de produção (*num_job*). O tratamento foi executado por meio da linguagem de programação *Python*, utilizando a biblioteca *Pandas* para manipulação dos dados e as bibliotecas *datetime* e

time para manipulação das datas. Dessa forma, foi possível obter a base de sequenciamento no formato apresentado na FIGURA 43.

FIGURA 43 – BASE DE DADOS *MS_TMP* NO FORMATO CORRETO UTILIZANDO A BASE FORNECIDA PELA PRODUZA.

In [4]: `ms_tmp`

Out[4]:

	Processo	Processo 2	Processo 3
OF			
657402001	1	0	0
683101001	1	0	0
687601001	1	0	0
705702001	1	0	0
705704001	1	0	0
...	...	...	...
763911001	1	0	0
763912001	1	0	0
766101001	1	0	0
766102001	1	0	0
766201001	1	0	0

247 rows × 3 columns

FONTE: O autor (2023).

É importante destacar que as possíveis máquinas dos roteiros da empresa foram redefinidas para números, conforme o dicionário {'SMD': 1, 'PTH': 2, 'INSPEÇÃO': 3, 'VERNIZ': 4}. A atribuição das máquinas como números é fundamental para o funcionamento do algoritmo, conforme mencionado anteriormente. Foi verificado e validado que as três "OFs" da base que possuem mais de uma máquina estavam de acordo, como mostra a FIGURA 44.

FIGURA 44 – VALIDAÇÃO DAS ORDENS DE PRODUÇÃO QUE POSSUEM MAIS DE UMA MÁQUINA PARA A BASE DE DADOS *MS_TMP*.

In [9]: `ms_tmp[ms_tmp['Processo 2']!=0]`

Out[9]:

	Processo	Processo 2	Processo 3
OF			
731802001	1	2	3
731804001	1	4	0
747901001	1	4	0

FONTE: O autor (2023).

Outro fato importante de se salientar é que não é relevante para o funcionamento do algoritmo o nome das colunas, e sim a maneira como as informações estão dispostas para cada uma delas, com uma ordem de produção por linha e respeitando a ordem sequencial. Durante as adaptações, o nome das colunas foi repetido com base na planilha da empresa para facilitar a manipulação dos dados. Dessa forma, foi possível obter a base de tempos de processamento no formato da FIGURA 45.

FIGURA 45 – BASE DE DADOS *PT_TMP* NO FORMATO CORRETO UTILIZANDO A BASE FORNECIDA PELA PRODUZA.

```
In [7]: pt_tmp
```

```
Out[7]:
```

	Tempo previsto em segundos	Tempo previsto em segundos 2	Tempo previsto em segundos 3
OF			
657402001	1890	0.0	0.0
683101001	93000	0.0	0.0
687601001	15750	0.0	0.0
705702001	31500	0.0	0.0
705704001	31500	0.0	0.0
...	...	...	...
763911001	252	0.0	0.0
763912001	63	0.0	0.0
766101001	630	0.0	0.0
766102001	80	0.0	0.0
766201001	63000	0.0	0.0

247 rows × 3 columns

Fonte: O autor (2023).

Nesse caso, também foi avaliado e validado como estavam dispostos os tempos de processamento das "OFs" que exigem mais de uma máquina, ilustrados na FIGURA 46. Além disso, foi verificado na planilha original se os valores estavam de fato de acordo com o tempo previsto em horas de cada máquina para essas ordens específicas, que são praticamente o fator motivador de toda a tratativa.

FIGURA 46 – VALIDAÇÃO DAS ORDENS DE PRODUÇÃO QUE POSSUEM MAIS DE UMA MÁQUINA PARA A BASE DE DADOS *PT_TMP*.

```
In [8]: pt_tmp[pt_tmp['Tempo previsto em segundos 2']!=0]
```

```
Out[8]:
```

	Tempo previsto em segundos	Tempo previsto em segundos 2	Tempo previsto em segundos 3
OF			
731802001	9300	7440.0	8370.0
731804001	9000	8100.0	0.0
747901001	28350	26100.0	0.0

Fonte: O autor (2023).

Dessa forma, foi obtida a base de prioridades de tempos para entrega, conforme ilustrado na FIGURA 47.

FIGURA 47 – BASE DE DADOS *JOB_PRIORITY_DUEDATE_TMP* ORIUNDA DA PLANILHA DA EMPRESA PRODUZA.

In [13]: <code>job_priority_duedate_tmp</code>		
Out[13]:		
	Prioridades	Data
OF		
657402001	110	4233600
683101001	191	10800000
687601001	152	6912000
705702001	116	4320000
705704001	117	4320000
...	...	...
763911001	167	7257600
763912001	168	7257600
766101001	106	3110400
766102001	105	3024000
766201001	156	6912000
247 rows × 2 columns		

FONTE: O autor (2023).

Como essa base representa valores únicos para cada ordem de produção que não dependem da quantidade de máquinas e sequenciamento, apenas foi validado que o número de linhas é igual às outras bases geradas. Para os tempos (*Data*), foi realizada uma breve validação de algumas "OFs", calculando a diferença de datas na planilha e gerando uma nova base, conforme ilustrado na FIGURA 48.

As três bases geradas foram ordenadas em ordem crescente em relação ao número da OF. Assim, garantiu-se que em todas as bases, as OFs foram posicionadas da mesma maneira, juntamente com suas respectivas informações. Isso é fundamental para a execução do algoritmo e o funcionamento do sistema de armazenamento de tempos para as máquinas e ordens de produção.

Para adaptar o algoritmo para o funcionamento com ordens de produção que possuem quantidades diferentes de máquinas, inicialmente foi criada uma lista de

frequências a partir da base de dados fornecida, utilizando a biblioteca *Numpy*. A lista contempla quantas vezes cada ordem de produção precisa ser agendada, de acordo com sua quantidade de máquinas. A lista é denominada *job_nof_sqdls* e foi gerada a partir da criação de uma matriz *job_nof_sqdls_array* utilizando o *Numpy*, onde é realizada a contagem das células não nulas de cada coluna para cada linha da base *ms_tmp*, respeitando a sequência de contagem conforme a ordem padrão dos índices utilizada até o momento. Posteriormente, a matriz é convertida em lista para facilitar a criação de estruturas de repetição com a variável. Após a criação da lista, é realizada a criação de uma função de restauração da população após a geração da população inicial.

FIGURA 48 – COMPARATIVO PARA VALIDAR PARTE DA BASE DE DADOS *JOB_PRIORITY_DUEDATE_TMP*. OBSERVA-SE COMO OS VALORES PARA DATA DA BASE E DIFERENÇA NA PLANILHA SÃO IGUAIS.

DIFERENÇA NAT FERRILHA OAS 1000000

Data		 PRODUZA			
OF		OF	Data	Data Hoje	Diferença
657402001	49	00657402001	30/03/23	09/02/23	49
		00683101001	14/06/23	09/02/23	125
683101001	125	00687601001	30/04/23	09/02/23	80
		00705702001	31/03/23	09/02/23	50
687601001	80	00705704001	31/03/23	09/02/23	50
		00705706001	31/03/23	09/02/23	50
705702001	50	00705708001	31/03/23	09/02/23	50
		00705710001	31/03/23	09/02/23	50
705704001	50	00706902001	02/02/23	09/02/23	-7
		00707001001	15/02/23	09/02/23	6
705706001	50				
705708001	50				
705710001	50				
706902001	-7				
707001001	6				

Fonte: O autor (2023).

A estrutura de repetição criada é dupla, onde itera-se por todas as soluções da população e, por cada alelo das soluções. A restauração é simples, pois os alelos já representam os índices (ordens). Toda vez que a contagem do alelo é inferior à sua contagem na lista, este é adicionado à solução. Caso a contagem do alelo seja superior ao seu valor na respectiva lista de frequências, este é eliminado da solução.

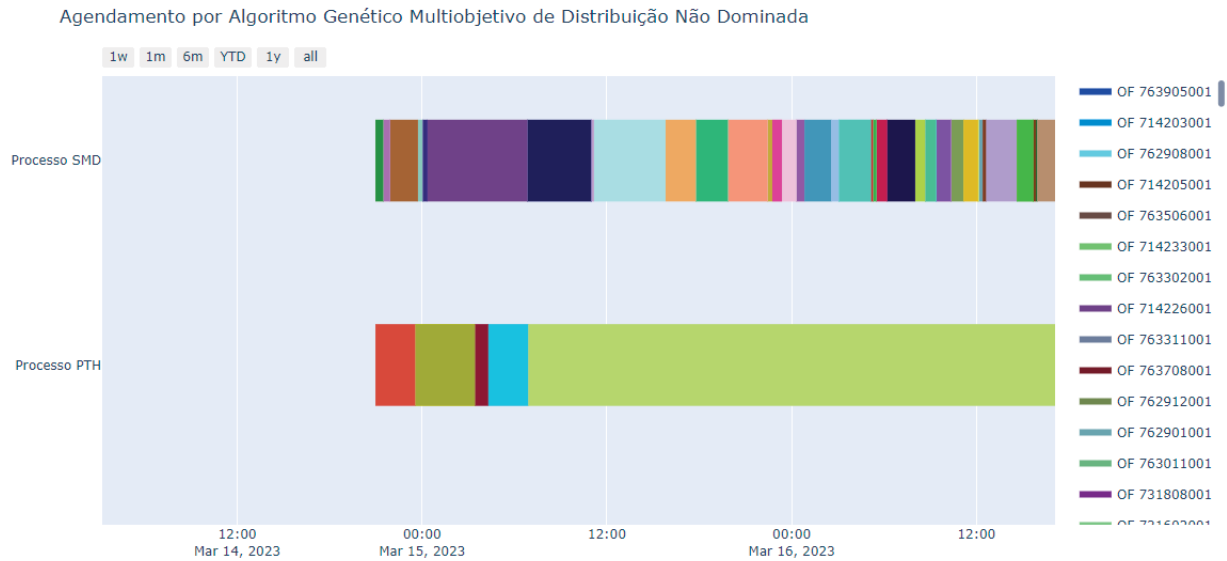
Concluída a correção para a etapa de geração da população inicial, foram direcionados esforços para a etapa da função de reparo, após o cruzamento de dois pontos. Na função de reparo, a lógica é mais complexa. No intuito de apenas realizar a troca dos alelos em excesso pelos alelos faltantes, sem afetar o posicionamento dos alelos das soluções filhas (*offspring_list*), são armazenados as contagens e o posicionamento das ordens de produção em excesso e faltantes. Com essas informações, o sistema realiza uma estrutura de repetição que possibilita a troca desses elementos. Anteriormente, a estrutura utilizava uma variável denominada *num_mc*, que representa o número de máquinas. No entanto, para diferentes ordens de produção, podem ser admitidas diferentes quantidades de máquinas para o problema prático. Assim, todas as ocorrências de *num_mc* na função de reparo foram substituídas pela lista *job_nof_schedules[x]*, onde o índice *x* é detalhado conforme a necessidade ou lógica de cada condicional ou sequência de repetição em que a variável *job_nof_schedules* substituiu o *num_mc*.

Através da verificação dos agendamentos gerados e das frequências de cada ordem nas listas, foi possível validar que as modificações necessárias para adaptar o algoritmo ao funcionamento com máquinas variáveis foram satisfatórias. Para a questão da função de reparo, foi necessária a validação das contagens de cada solução.

Após a conclusão das modificações, foi possível gerar uma resposta adaptada do algoritmo *NSGA-II* proposto por Wu (2018) para a base de dados da empresa *Produza*, com um agendamento otimizado das 251 ordens de produção de acordo com a base de dados. Como mencionado, a resposta consiste inicialmente de um conjunto de 20 soluções finais para o agendamento, com valores iguais ou diferentes para as funções objetivo. Além disso, é apresentada uma lista contendo o valor atingido para cada função fitness das respectivas funções, no formato [*twet*, *makespan*]. Devido à extensão das respostas, apresentaremos apenas a primeira lista de agendamentos e os valores das funções objetivo para todas as soluções da melhor fronteira:

[[217, 51, 169, 110, 114, 18, 24, 89, 125, 154, 208, 48, 71, 119, 50, 109, 212, 210, 219, 131, 141, 14, 202, 124, 98, 191, 132, 74, 113, 144, 36, 105, 107, 194, 130, 129, 60, 232, 69, 101, 135, 37, 171, 111, 179, 181, 20, 61, 106, 82, 56, 153, 118, 164, 7, 158, 86, 127, 128, 55, 215, 96, 117, 187, 57, 16, 168, 47, 170, 72, 94, 190, 183, 38, 155, 123, 221, 151, 5, 140, 59, 108, 161, 233, 196, 32, 228, 126, 78, 52, 42, 112, 85, 17, 195, 159, 102, 180, 87, 58, 133, 80, 224, 121, 148, 34, 231, 29, 201, 206, 138, 186, 25, 93, 182, 35, 70, 139, 84, 172, 31, 49, 146, 136, 157, 6, 81, 197, 90, 115, 163, 64, 19, 3, 235, 192, 92, 227, 62, 26, 160, 33, 178, 0, 165, 91, 205, 223, 149, 150, 166, 142, 53, 103, 27, 30, 199, 226, 66, 234, 134, 39, 245, 162, 44, 239,

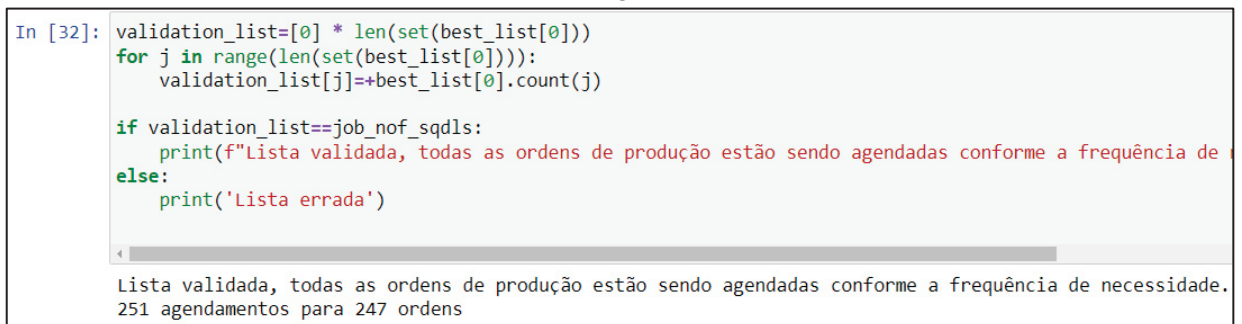
FIGURA 50 – VISÃO AMPLIADA DO CALENDÁRIO DE GANTT, PARA FACILITAR A IDENTIFICAÇÃO DAS ORDENS.



Fonte: O autor (2023).

Para validar se a solução contempla todos os agendamentos necessários para todas as ordens, foi criado um código de validação, conforme apresentado na FIGURA 51, que confirma se estão sendo contemplados todos os agendamentos necessários para todas as ordens. Foi criada uma lista "*validation_list*" que calcula a quantidade de ordens de produção agendadas na resposta, que foi comparada com a lista utilizada para adequar o algoritmo, "*job_nof_schedules*". Caso essas variáveis sejam iguais, o algoritmo retorna se a lista foi validada ou não.

FIGURA 51 – VALIDAÇÃO DA FREQUÊNCIA DE AGENDAMENTO DE CADA ORDEM NA RESPOSTA.



Fonte: O autor (2023).

Em termos da funcionalidade do modelo, devem ser avaliadas posteriormente manipulações dos dados de entrada do algoritmo, bem como o impacto da utilização de diferentes funções fitness. Foi observado, em um primeiro momento, que as ordens atrasadas não estão sendo agendadas com a prioridade que deveriam. Assim, foi

eliminada a parcela de cálculo da função *TWET* referente à antecipação das ordens, considerando apenas a prioridade multiplicada pelo atraso para efeito de teste. A alteração já apresentou melhorias, com ordens atrasadas sendo agendadas com maior prioridade pelo sistema, podendo ser um ponto de partida para futuras análises e explorações das funções objetivo do algoritmo.

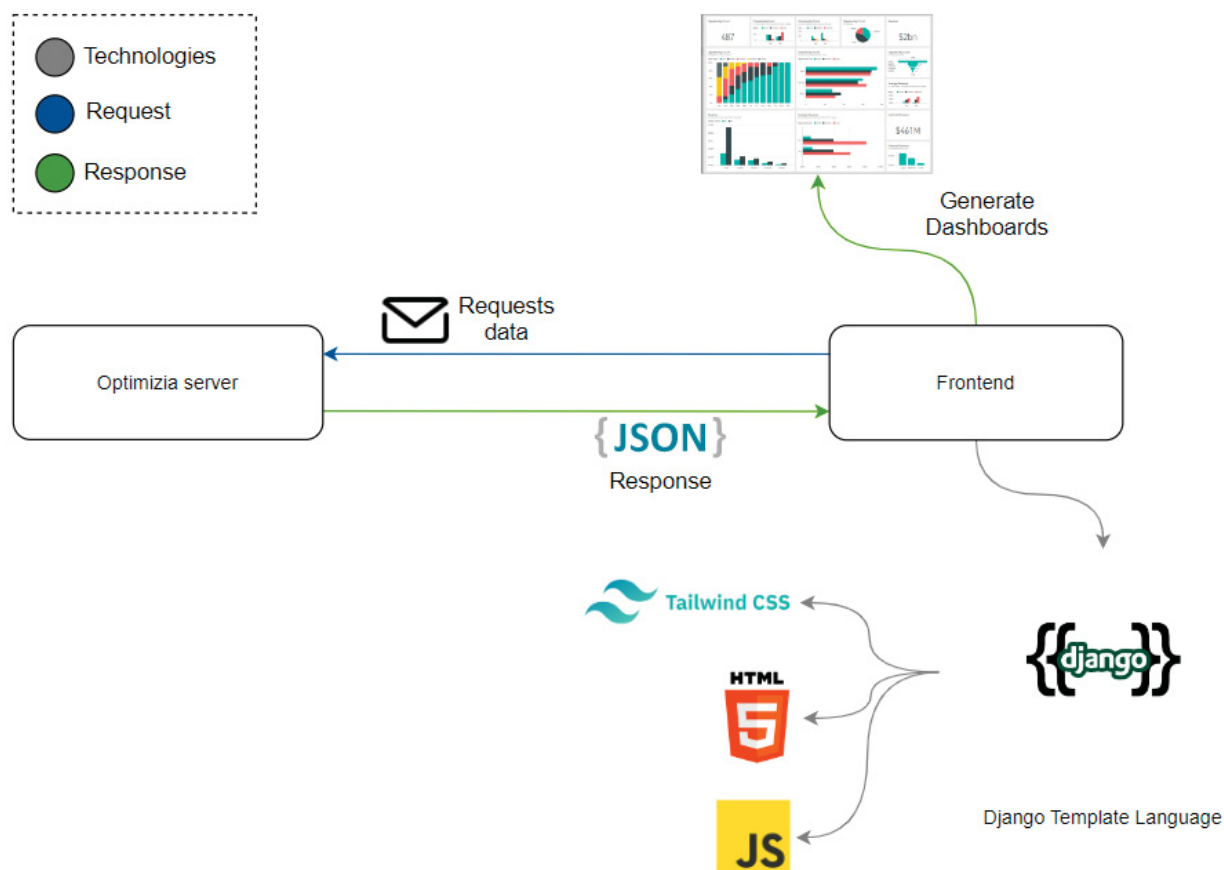
4.3.4.3. Desenvolvimento da Solução - Front-end

Para que o *front-end* esteja funcionando no servidor utilizando o *framework Django*, é necessário que o desenvolvedor realize diversas atividades. Inicialmente, é importante que ele defina quais serão as funcionalidades que estarão disponíveis na interface, a fim de que a estrutura da página seja projetada de forma adequada para receber essas informações.

O *front-end* é responsável por interpretar as informações que são enviadas pelo *back-end* e apresentá-las de forma amigável e fácil de usar para o usuário final. Assim, para gerar uma interface eficiente, é necessário seguir uma série de etapas, desde a requisição dos dados e métricas até a exibição das informações de forma visual para o usuário. Nesse contexto, a requisição dos dados e métricas é uma das etapas mais importantes, pois é nela que se inicia a manipulação dos dados e a geração das informações que serão exibidas no *front-end*.

O processo de solicitação de dados pelo *front-end* para geração de métricas é uma atividade fundamental para o funcionamento do *dashboard*. O *front-end* envia uma solicitação de dados para o servidor da solução, informando quais informações são necessárias para gerar as métricas solicitadas. Em seguida, o servidor coleta as informações da base de dados e as manipula para gerar as métricas. Este processo envolve a aplicação do algoritmo processado, que foi implementado na linguagem de programação *Python*.

Após a geração das métricas, os dados são enviados em resposta para o *front-end* no formato *JSON*. Este formato é amplamente utilizado para a troca de informações em aplicações web, devido à sua simplicidade e flexibilidade. O *front-end*, por sua vez, utiliza os dados recebidos para exibir as informações no *dashboard*. Esta arquitetura pode ser observada na FIGURA 52.

FIGURA 52 – ARQUITETURA DO *FRONT-END*.

Fonte: O autor (2023).

A arquitetura utilizada neste processo é essencial para garantir o correto funcionamento do dashboard. Ela permite que o *front-end* e o servidor se comuniquem de forma eficiente, trocando informações de forma rápida e segura. Além disso, a utilização de uma arquitetura modularizada permite que diferentes partes do sistema possam ser atualizadas e mantidas de forma independente, facilitando a manutenção e evolução da solução.

Com base na estruturação da página, a tecnologia *HTML5* foi utilizada para criar os elementos da interface, tais como botões, campos de texto, imagens e outros elementos que serão exibidos ao usuário. Esses elementos serão organizados em uma página que apresente um *layout* agradável e intuitivo, que facilite a interação do usuário com a aplicação.

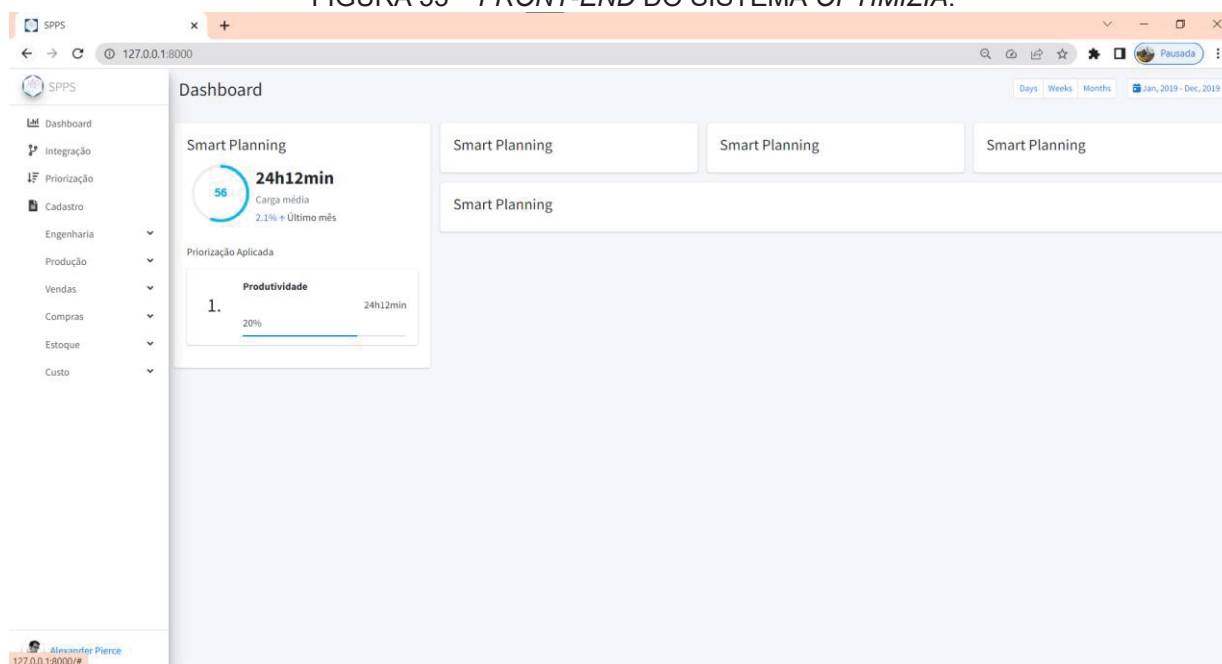
Para estilizar a interface visual do *front-end*, o *framework Tailwind CSS* foi utilizado. Essa tecnologia oferece diversas opções de estilização, permitindo que a página seja customizada de acordo com as necessidades da aplicação. Com o *Tailwind CSS*, é possível definir cores, fontes, tamanhos de texto, margens e outros atributos visuais que ajudarão a criar uma interface atraente e fácil de usar.

Por fim, o *JavaScript* foi utilizado para adicionar interatividade à interface. Essa tecnologia permite que as informações sejam atualizadas e manipuladas em tempo real, sem que seja necessário recarregar a página. O *JavaScript* pode ser utilizado para criar animações, adicionar validação de formulários e outras funcionalidades que aumentem a usabilidade da aplicação.

Após o desenvolvimento do *front-end*, é necessário realizar os testes necessários para garantir que a interface está funcionando corretamente e que o usuário final terá uma experiência positiva ao utilizá-la.

Com o *front-end* pronto e testado, foi realizado o *deploy* no servidor. Para isso, foi necessário configurar o ambiente do servidor para que o *Django*, o *Tailwind CSS*, o *HTML5* e o *JavaScript* possam ser executados. Foi necessário configurar o servidor *web*, o banco de dados e outras ferramentas necessárias para o correto funcionamento do *front-end*, conforme é possível visualizar na FIGURA 53.

FIGURA 53 – FRONT-END DO SISTEMA OPTIMIZIA.



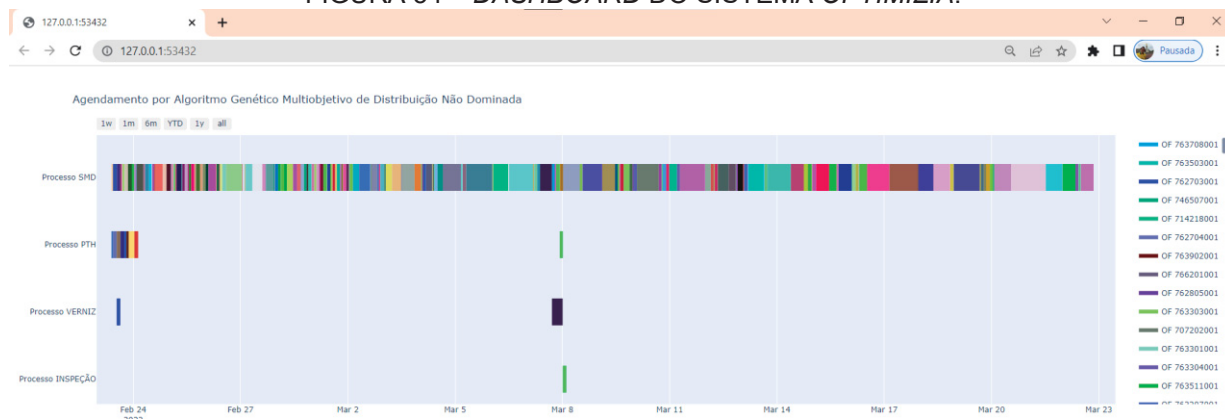
Fonte: O autor (2023).

Já o *dashboard* pode apresentar diferentes tipos de gráficos, tabelas, gráficos de barras, mapas, entre outros recursos visuais, que ajudam a identificar tendências e padrões de maneira rápida e eficiente.

Um dashboard eficiente é aquele que apresenta as informações de forma clara e organizada, permitindo que o usuário possa compreender e analisar os dados de

maneira fácil e rápida. Desta forma, a concepção do *dashboard* foi definida na etapa de *design* e seu desenvolvimento pode ser contemplado na FIGURA 54.

FIGURA 54 – DASHBOARD DO SISTEMA OPTIMIZIA.



Fonte: O autor (2023).

Por fim, é importante garantir que a interface do usuário esteja sempre disponível e funcionando corretamente no servidor. É necessário monitorar a interface para identificar possíveis problemas e realizar atualizações e manutenções periódicas para garantir que ela continue funcionando adequadamente.

4.4. ETAPA 4 – AVALIAÇÃO

Na fase de análise, Santos (2018) argumenta que o método *Design Science* busca tanto a validade científica, que diz respeito ao rigor na concepção e condução da pesquisa, quanto a validade pragmática, que se refere à eficácia e efetividade das soluções propostas. Isso significa que o método *Design Science* não se preocupa apenas em produzir conhecimento, mas também em gerar soluções práticas para problemas reais.

No desenvolvimento desta pesquisa, a validade científica foi alcançada por meio de uma sólida revisão sistemática da literatura, realizada na fase 1. A revisão foi utilizada ao longo de todo o projeto, servindo como referencial teórico e fornecendo informações relevantes para o direcionamento do trabalho e da pesquisa. A revisão sistemática é uma técnica amplamente reconhecida na academia por sua capacidade de fornecer uma síntese crítica e objetiva do conhecimento disponível sobre um determinado assunto.

Por sua vez, a validade pragmática foi alcançada por meio do desenvolvimento do *framework* do produto tecnológico *OptimizIA*, utilizando o Algoritmo Genético de Classificação Não Dominada II (*NSGA-II*), e a aplicação desse algoritmo na simulação da programação de produção da empresa Produza, nas fases 2, 3 e 4 deste projeto. O *framework* desenvolvido permitiu a avaliação da solução para a geração da programação da fábrica de uma maneira mais automatizada, do que atualmente, já que hoje são utilizadas planilhas e conhecimento adquirido dos gestores.

Assim, fica claro que a utilização do *método Design Science* na presente pesquisa possibilitou a integração entre a pesquisa acadêmica e as necessidades práticas da empresa, resultando em uma solução eficaz e efetiva para o problema proposto. O *framework* desenvolvido pode ser considerado como uma importante contribuição para a área de gestão da produção, ao mesmo tempo em que representa um avanço na aplicação de técnicas de inteligência artificial na resolução de problemas empresariais.

4.5. ETAPA 5 – CONCLUSÕES

Santos (2018) recomenda que as conclusões se prendam ao problema central da pesquisa e seus correspondentes objetivos, desta forma, como o objetivo da pesquisa foi desenvolver um *framework* de um sistema avançado de planejamento de produção para o ambiente de manufatura, utilizando sistemas computacionais de inteligência artificial e aplicando conceitos da Indústria 4.0. Pode-se afirmar que o *framework* sugerido foi desenvolvido utilizando algoritmo de inteligência artificial *NSGA II*, podendo assim, ajudar a resolver os problemas levantados nas primeiras fases da metodologia, dando flexibilidade, reduzindo a dependência humana e de sistemas auxiliares no planejamento da produção, possibilitando uma inteligência que considere outros parâmetros e não apenas restrições, e que possibilite aos sistemas avançados de planejamento da produção acompanhar os últimos avanços relacionados à transformação digital e à indústria 4.0.

Os objetivos específicos estão descritos abaixo com os resultados alcançados.

- 1) Identificar na literatura como os sistemas avançados de planejamento e programação da produção (APS) são abordados, sendo alcançado com a Revisão Bibliográfica Sistemática.

- 2) Selecionar o algoritmo de Inteligência Artificial mais apropriado para solucionar problemas de *APS*, com a aplicação do algoritmo genético multiobjetivo *NSGA II*.
- 3) Projetar o *framework* do sistema de *APS* com o desenvolvimento do *MVP* do produto comercial *OPTimizIA*.
- 4) Avaliar a aplicabilidade do algoritmo em um problema real de *APS*, com a aplicação do algoritmo no estudo de caso na empresa *Produza*.

5. RESULTADOS E DISCUSSÃO

O planejamento e a programação avançada de produção, conhecidos pela sigla *APS*, são essenciais para a gestão eficiente e competitiva de empresas em diversos setores. Diante da importância desses sistemas, muitos estudos vêm sendo realizados para aprimorar e desenvolver novas soluções. Nesse contexto, a revisão sistemática bibliográfica se destaca como uma técnica que permite a coleta e análise de estudos relevantes, contribuindo para a evolução do conhecimento na área.

Além disso, o desenvolvimento de sistemas *APS* requer uma abordagem metodológica que contemple as particularidades do processo produtivo das empresas, bem como as necessidades e expectativas dos usuários. A *design science review* é uma metodologia que tem sido aplicada com sucesso para o desenvolvimento de sistemas em diversas áreas, incluindo a produção e logística. Essa abordagem envolve a criação de soluções inovadoras e a validação prática dos resultados, com foco na geração de valor para os usuários e a organização.

Nesse contexto, o presente capítulo apresenta o resultado de um estudo que envolveu a revisão sistemática bibliográfica sobre sistemas *APS*, a aplicação da metodologia *design science review* no desenvolvimento de um *software* para gestão de produção e a implementação de algoritmos inteligentes para otimizar o processo produtivo. O objetivo é apresentar uma solução tecnológica inovadora e eficiente para a gestão de produção, que possa ser aplicada em empresas de diferentes setores.

Para isso, são detalhados os principais resultados da revisão sistemática bibliográfica, seguidos pelo processo de desenvolvimento do *software APS* e a aplicação da metodologia *design science review*. Em seguida, será apresentada a solução tecnológica desenvolvida, destacando os algoritmos inteligentes implementados e suas contribuições para a otimização do processo produtivo. Por fim, são discutidas as conclusões do estudo e as perspectivas futuras para a evolução dos sistemas *APS*.

5.1. REVISÃO BIBLIOGRÁFICA SISTEMÁTICA (RBS)

A primeira etapa do projeto foi a realização de uma revisão sistemática da literatura, a fim de localizar e utilizar como base as pesquisas e o conhecimento científico já existente sobre o *APS* e dentro dos conceitos de Indústria 4.0.

Para este estudo, o método de revisão bibliográfica utilizado para atingir os objetivos propostos foi o *Knowledge-Constructivist Development Process (Proknow-C)* (ENSSLIN e ENSSLIN, 2007; TASCA *et al.*, 2010). O processo consiste em quatro etapas: (a) seleção de um portfólio de artigos sobre o tema de pesquisa; (b) análise bibliométrica do portfólio; (c) análise sistêmica; e, (d) definição da questão de pesquisa e objetivo da pesquisa (ENSSLIN, ENSSLIN e PINTO, 2013).

A formação de um portfólio de artigos permite ao pesquisador reunir uma gama de artigos relacionados ao tema da pesquisa, alinhados de acordo com sua percepção e com os limites impostos. O produto final dessa etapa é um conjunto de artigos que os pesquisadores consideram relevantes, em consonância com sua pesquisa, então denominado Portfólio Bibliográfico - PB (ENSSLIN, ENSSLIN e PINTO, 2013).

No caso do presente estudo, conforme Vieira, Deschamps e Valle (2021) e apresentado na FIGURA 18, foi definido o eixo de pesquisa sistemas avançados de planejamento e programação da produção. Em seguida, a primeira fase dessa seleção dos artigos brutos ocorre com a definição das palavras-chave para o eixo citado. Duas palavras-chave foram definidas “Sistema de Planejamento Avançado” e “Planejamento e Programação Avançada”. A próxima etapa centrou-se na definição das bases de dados disponibilizadas no Portal de Periódicos da CAPES, a partir do qual foram selecionadas as bases de dados *Web of Science* e *Scopus*, que indexam uma gama de periódicos científicos mais alinhada ao tema pesquisado. A partir da definição das bases, inicia-se um processo de busca, utilizando as combinações de palavras-chave definidas por eixo de pesquisa, restringindo-se aos campos de títulos dos artigos (título do artigo), palavras-chave e resumos (resumos), um período de 20 anos antes da pesquisa (2001 a 2021), de trabalhos publicados apenas em periódicos.

Com as buscas realizadas nas bases de dados de periódicos utilizando as palavras-chave definidas pelos autores, foi alcançado um total de 649 trabalhos publicados, que compõem o Banco de Artigos Brutos. Para a reunião dos trabalhos e composição dos Bancos de Artigos, foi utilizado o *Endnote* como gestor bibliográfico. Com o portfólio bruto combinado, a aderência das palavras-chave é testada. Para tanto, foram escolhidos três artigos de forma aleatória, com o objetivo de identificar ou não a necessidade de inclusão de novas palavras-chave. Porém, na execução desses procedimentos, concluiu-se que não seria necessária a inclusão de novas palavras-chave, o que indicava o alinhamento desses artigos com o tema da pesquisa.

Assim, partimos de uma base de 649 artigos e, com a utilização do gerenciador bibliográfico, foi possível identificar e excluir artigos redundantes, o que resultou na exclusão de 194 artigos, restando assim 455 artigos para a análise do alinhamento dos títulos. Com a leitura de todos os títulos, chegamos a 140 artigos com títulos alinhados ao tema da pesquisa. A partir daí, foi elaborada uma planilha para identificar o grau de reconhecimento científico das publicações. Para tanto, foi novamente consultado o site do *Scopus* e *Web Of Science*, a fim de aumentar o número de citações de cada uma das obras.

Após a fase de consulta de citações, 90% das citações de obras foram consideradas representativas; e 10% eram artigos com menos de 6 citações. Em números absolutos, 46 artigos ou 90% das citações passaram a fazer parte do repositório de títulos alinhados e cientificamente reconhecidos, tratado para efeito de estudo como repositório *k*. Os outros 94 artigos, ou 10% das citações, foram separados em outro repositório, denominado repositório *p*, classificado como títulos alinhados com menor reconhecimento científico ou com reconhecimento ainda não confirmado. Em seguida, foram lidos os resumos dos 46 artigos do repositório *p* para verificar o alinhamento do conteúdo com o tema. Destes, 29 se enquadram no tema e foram classificados como parte de um novo repositório, denominado repositório *a*, excluindo-se os demais trabalhos não enquadrados do repositório *p*. A partir daí, foram identificados os autores dos artigos, formando-se assim um Banco de Autores (BA).

Como parte do processo, também foi realizada uma análise dos artigos do repositório *k*, a fim de identificar outros artigos com potencial para compor o portfólio bibliográfico. Dos 94 artigos do repositório *p*, 12 foram publicados entre 2018 e 2021, alguns dos quais sem reconhecimento científico, mas com potencial para tirar partido do seu conteúdo por serem recentes. Dos outros 82 artigos, foi feita uma análise de autores comparando com os autores do Banco de Autores (BA), desta análise foram recuperados 17 artigos com potencial de reconhecimento científico, os quais foram adicionados aos outros 12 artigos, formando o repositório *b* com 29 artigos.

Em seguida, seus resumos foram lidos, eliminando 3 artigos, restando 26 artigos que foram adicionados aos 29 do repositório *a*, formando o repositório *c*, totalizando 55 artigos. Na fase posterior à filtragem, foi verificada a disponibilidade dos artigos na íntegra. Para efeito desta pesquisa, um fator limitante dos pesquisadores foi a disponibilização gratuita dos artigos na íntegra. Assim, dos 55 artigos, 19 não estavam disponíveis na íntegra. Em seguida, iniciamos a leitura dos artigos completos

a fim de confirmar o alinhamento com a temática de pesquisa em questão, dos quais 36 foram considerados alinhados com o tema da pesquisa, finalizando aqui o processo de filtragem dos artigos.

Este estudo gerou um artigo científico que foi publicado no *The 28th Conference on Transdisciplinary Engineering* (TE2021).

5.2. DESIGN SCIENCE RESEARCH (DSR)

A segunda etapa do projeto foi estudar o método para a realização da pesquisa científica e desenvolvimento do trabalho, a fim de sistematizar e entender cada uma das cinco fases do método na condução do estudo.

Desta forma, de acordo com Vieira, De Aguiar e Deschamps (2021) as cinco fases sugeridas por Manson (2006) foram seguidas rigorosamente para sistematizar a condução do *DSR*. O problema de pesquisa foi identificado a partir da revisão sistemática da literatura e confirmado durante a execução do estudo. Um conjunto de artefatos foi elaborado para delimitar os objetivos. Alternativas foram geradas para solucionar os problemas e objetivos, materializados com o *mock-up* da solução na fase 2. Na fase 3, foi desenvolvido um protótipo de alta fidelidade, com foco na aquisição, processamento e disposição de informações e dados. Na fase 4, foi realizada uma análise com base na validade científica e na validade pragmática e, por fim, a última fase foi apresentada com base na conclusão e nos resultados.

O foco deste trabalho de pesquisa foi a aplicação teórica da metodologia *DSR* ao desenvolvimento de um sistema *APS*, seguindo todas as etapas, mas os pesquisadores sugerem que o ciclo seja refeito em pesquisas futuras, com foco em pontos não abordados neste estudo, como estrutura de integração com outros sistemas, desenvolvimento do algoritmo de inteligência artificial e que este produto seja realmente desenvolvido comercialmente, utilizando uma análise de mercado, tanto na fase de definição do artefato, quanto na análise de usabilidade e atendimento aos objetivos do sistema e que o foco da solução está voltado para a evolução dos sistemas *APS* dentro dos conceitos de transformação digital e indústria 4.0.

Este estudo gerou um artigo científico que foi publicado no *I Latin American Congress of Applied Technologies* (2021).

5.3. PRODUTO TECNOLÓGICO

O sistema *OPTimizIA* foi desenvolvido com a ideia de ser um *MVP* (*Minimum Viable Product*) que utiliza apenas duas funções objetivas de otimização, o *makespan* e o *TWET*. Essas funções foram escolhidas para atender às necessidades específicas da indústria, mas a ideia é que outras funções de otimização sejam incorporadas no futuro, de acordo com as necessidades dos usuários.

A escolha dessas duas funções objetivas de otimização permite que o *OPTimizIA* possa ser facilmente utilizado por empresas de diferentes setores e tamanhos, garantindo uma otimização eficiente e personalizada para cada caso.

O sistema *OPTimizIA* passou por diversas fases de desenvolvimento, incluindo o *design*, a implementação do *back-end* e *front-end*, além da implementação do Algoritmo Genético de Classificação Não Dominada II (*NSGA-II*) para a otimização das funções objetivas.

A fase de *design* foi crucial, pois permitiu a definição dos requisitos e especificações do sistema, bem como a identificação das funcionalidades que seriam implementadas. Durante essa fase, foram realizados diversos estudos de caso e pesquisas de mercado para garantir que o *OPTimizIA* atendesse às necessidades da indústria de forma eficiente e personalizada.

Em seguida, foi realizada a implementação do *back-end* e *front-end*, que permitiu a criação da interface do usuário e a integração com os sistemas de produção. O *back-end* do *OPTimizIA* foi desenvolvido para garantir a segurança e a escalabilidade do sistema, além de permitir a implementação de novas funcionalidades e a integração com outros sistemas. Já o *front-end* foi projetado para garantir uma interface intuitiva e amigável, permitindo que os usuários possam visualizar e gerenciar os dados de produção de forma eficiente.

Por fim, o Algoritmo Genético de Classificação Não Dominada II (*NSGA-II*) foi implementado para permitir a otimização da produção de forma automatizada. Essa técnica de otimização permite encontrar soluções que equilibrem diversas variáveis, como a minimização dos custos, a maximização da produção e a otimização da utilização dos recursos.

O sistema *OPTimizIA* se apresenta como uma solução completa e eficiente para a gestão da produção, permitindo que as empresas possam obter benefícios imediatos em termos de eficiência, qualidade e redução de custos. Além disso, o *OPTimizIA* se

apresenta como uma solução flexível e adaptável, que pode ser facilmente personalizada de acordo com as necessidades dos usuários e das empresas. Com o tempo, novas funções de otimização podem ser adicionadas, tornando o sistema ainda mais completo e eficiente para a gestão de operações industriais

6. CONSIDERAÇÕES FINAIS

Através do presente trabalho, foi desenvolvido e apresentado um *framework* de um sistema *APS* com aplicação comercial. Para isso, foram seguidas diversas etapas com o intuito de garantir o sucesso deste estudo e trabalho.

A metodologia adotada para o desenvolvimento foi a *DSR*, que foi rigorosamente seguida. A primeira etapa da metodologia resultou na publicação de um artigo sobre a Revisão Sistemática da Bibliografia (RSB), que auxiliou no embasamento teórico deste projeto.

Uma publicação sobre a *DSR* parcial também foi realizada, porém, não se focou no algoritmo inteligente e nem na solução final comercial. Ambos são elementos importantes para este trabalho, especialmente a solução final, que consistiu no desenvolvimento de um produto minimamente viável, incluindo o *design*, o *front-end* e o *back-end*.

O algoritmo genético multiobjetivo *NSGA II* foi selecionado e aplicado como algoritmo inteligente para a solução final, focando nas funções objetivas de *makespan* e *TWET*.

Por fim, a solução *OPTimizIA* foi aplicada na empresa *Produza*, sendo testada com a simulação de suas ordens de produção, permitindo a sequenciação das mesmas ao longo do tempo. Este *MVP* focou apenas na aplicabilidade da solução em um caso real, porém não foram analisados os ganhos financeiros que esta ferramenta pode trazer ao processo produtivo da empresa *Produza*.

Este trabalho contribuiu para a pesquisa e aplicação de técnicas avançadas de inteligência artificial, bem como para a criação de uma solução comercial viável. A partir daqui, novas pesquisas e melhorias podem ser implementadas, a fim de aprimorar ainda mais o sistema *APS* desenvolvido.

6.1. DELIMITAÇÕES DO TRABALHOS

Embora a solução *OPTimizIA* tenha sido desenvolvida seguindo rigorosamente a metodologia *DSR* e tenha alcançado resultados satisfatórios na simulação das ordens de produção da empresa *Produza*, é importante destacar algumas limitações que precisam ser consideradas. Primeiramente, a falta de conhecimento em programação de *softwares* e desenvolvimento de algoritmos de inteligência artificial exigiu a busca

por ajuda de profissionais, o que pode ter impactado na implementação da solução. Além disso, a falta de apoio de alguma organização para incluir o produto tecnológico em seu portfólio pode limitar sua adoção pelo mercado.

A aplicação desta solução na empresa *Produza* focou apenas se era possível programar ou não utilizando uma inteligência artificial com foco em objetivos das funções, evidenciando a falta de levantamento de ganhos financeiros com a aplicação da solução, comparando os cenários gerados com a execução real.

Outra limitação relevante é o foco em apenas duas funções objetivas, o *makespan* e o *TWET*, deixando de lado outros possíveis objetivos relevantes para a indústria, como a minimização de estoques ou a maximização da utilização dos recursos. Uma possível escolha para melhorar o trabalho seria incluir outras funções objetivas na modelagem do problema.

Adicionalmente, a falta de testes e experimentos práticos em outras empresas limita a generalização da solução *OptimizIA* para diferentes contextos e situações. Seria interessante explorar outras técnicas de inteligência artificial, como redes neurais ou algoritmos de aprendizado por reforço, para comparar com os resultados obtidos com o algoritmo genético.

Diante dessas limitações, é importante reconhecer que um ambiente de desenvolvimento com profissionais com conhecimento em *design*, programação de *software* e ciência de dados, e a exploração de diferentes algoritmos de inteligência artificial poderiam aumentar a robustez da solução *OptimizIA* e ampliar suas possibilidades de aplicação em diferentes tipos de indústrias manufatureiras.

6.2. SUGESTÕES DE TRABALHOS FUTUROS

À medida que um estudo é realizado, muitas vezes surgem novas possibilidades de pesquisa que não puderam ser exploradas completamente durante o período de desenvolvimento do projeto. Com base nisso, é comum surgirem sugestões de trabalhos futuros que podem ser realizados para aprimorar e expandir o conhecimento gerado pelo estudo original.

Nesse sentido, o presente trabalho apresentou um *MVP* de uma solução comercial baseada em um *framework* de um sistema *APS*, desenvolvido a partir da aplicação da metodologia *DSR* e do algoritmo genético multiobjetivo *NSGA-II*. Diante

disso, algumas possibilidades de trabalhos futuros surgem, com o objetivo de aprimorar e ampliar os resultados obtidos no estudo original.

- 1) Ampliação do escopo do algoritmo inteligente: o estudo atual focou em apenas duas funções objetivas, o *makespan* e o *TWET*. Futuros trabalhos podem explorar a inclusão de outras funções objetivas, tornando o algoritmo mais abrangente e eficaz.
- 2) Utilização de outras técnicas de inteligência artificial, como redes neurais ou algoritmos de aprendizado por reforço, para comparar com os resultados obtidos com o algoritmo genético.
- 3) Validação da solução comercial: embora a solução tenha sido testada em simulações, seria importante validar sua eficácia em um ambiente real de produção, em parceria com outras empresas do setor.
- 4) Implementação de melhorias na interface do usuário: embora o *front-end* da solução tenha sido desenvolvido, ainda há espaço para melhorias na usabilidade e na interface do usuário, tornando-a mais amigável e intuitiva.
- 5) Aplicação da solução em outras empresas e setores: o estudo atual focou na aplicação da solução em uma única empresa do setor industrial. Futuros trabalhos podem explorar a aplicação da solução em outras empresas e setores, avaliando sua eficácia em diferentes contextos.
- 6) Comparação com outras soluções existentes: seria interessante comparar a eficácia da solução desenvolvida neste estudo com outras soluções existentes no mercado, avaliando seu desempenho e identificando possíveis limitações e oportunidades de melhoria.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARICA, Emrah; POWELL, Daryl J. A framework for ICT-enabled real-time production planning and control. **Advances in Manufacturing**, v. 2, n. 2, p. 158-164, 2014.
- BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. (Eds.). **Handbook of evolutionary computation**. Oxford University Press, 2000.
- BAKER, R.; SCUDDER, G. D. Sequencing with earliness and tardiness penalties: A review. **Operation Research**, 38 (1), 22–36, 1990.
- BILLINGTON, P. J.; MCCLAIN, J. D.; THOMAS, L. J. Mathematical programming approaches to capacity-constrained mrp systems: review, formulation and problem reduction. **Management Science**, 29 (10): 1126-1141, 1983.
- BISKUP, D.; JAHNKE, H. Common due date assignment for scheduling on a single machine with jointly reducible processing times. **International Journal of Production Economy**, 69, 317-322, 2001.
- BUXTON, B. **Sketching user experiences: getting the design right and the right design**. Morgan Kaufmann, 2007.
- CHEN, K.; JI, P. A mixed integer-programming model for advanced planning and scheduling (APS). **European Journal of Operational Research** 181 (1): 515-522, 2007.
- CHEN, K.; JI, P. A genetic algorithm for dynamic advanced planning and scheduling (DAPS) with a frozen interval. **Expert Systems with Applications**, 33 (4), 1004–1010, 2007.
- CHEN, K.; JI, P.; WANG, Q. A case study for advanced planning and scheduling (APS). **Journal of Systems Science and Systems Engineering**, 20(4): 460-474, 2011.
- CHENG, T. C. E.; GUPTA, M. C. Survey of scheduling research involving due date determination decisions. **European Journal of Operational Research**, 38 (2), 156–166, 1989.
- COOPER, A. **About face 2.0: the essentials of interaction design**. John Wiley & Sons, 1999.
- CORRÊA, L. H.; NOGUEIRA, I. G.; CAON, M. **Planejamento, Programação e Controle da Produção - MRP II / ERP**, 6ª edição. São Paulo: Atlas, 2019.
- COX, J.F.; BLACKSTONE, J. H. **APICS Dictionary**. APICS, VA, 2002.
- DAVID, F.; PIERREVAL, H.; CAUX, C. Advanced planning and scheduling systems in aluminium conversion industry. **International Journal of Computer Integrated Manufacturing**, 19 (7), 705–715, 2006.
- DE MAN, J. C.; STRANDHAGEN, J. O. Spreadsheet Application still dominates Enterprise Resource Planning and Advanced Planning Systems. **IFAC-PapersOnLine** 51(11): 1224-1229, 2018.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, Vol.6, No.2. IEEE Computational Intelligence Society, 2002.

DRESCH, *et al.* **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Bookman Editora. 2015.

ENSSLIN, L.; ENSSLIN, S. R. **Orientações para elaboração dos artigos científicos do LabMCDA-C [Apostila da disciplina Avaliação de Desempenho do Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina]**. Florianópolis. UFSC, 2007.

ENSSLIN, Leonardo; ENSSLIN, Sandra Rolim; PINTO, Hugo de Moraes. Processo de investigação e análise bibliométrica: avaliação da qualidade dos serviços bancários. **Rev. adm. contemp.** [online], vol.17, n.3, pp.325-349. 2013.

FISCHER, M. **Available-to-promise: Aufgaben und Verfahren im Rahmen des Supply Chain Management**, Regensburg, 2001.

G2. **Best Advanced Planning and Scheduling (APS) Software**. Disponível em: < <https://www.g2.com/categories/advanced-planning-and-scheduling-aps> > Acesso em: 03 jan. 2023.

GEN, M.; CHENG, R. **Genetic algorithms and engineering design**, New York: Wiley. 1997.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Addison-Wesley, 1989.

GORDON, V.; PROTH, J. M.; CHU, C. B. A survey of the state of-the-art of common due date assignment and scheduling research. **European Journal of Operational Research**, 139 (1), 1–25, 2002.

GUPTA, J. N. D.; NEPPALLI, V. R.; WERNER, F. Minimizing total flow time in a two-machine flowshop problem with minimum makespan. **International Journal of Production Economy**, 69, 323-338, 2001.

HANKINS, S. L.; WYSK, R. A.; FOX, K. R. Using a CATS database for alternative machine loading. **Journal of Manufacturing Systems**, 3, 115-120, 1984.

HASTINGS, N. A. J.; MARSHALL, P.; WILLIS, R. J. Schedule based MRP: An integrated approach to production scheduling and material requirements planning. **Journal of the Operational Research Society**, 3 (11), 1021–1029, 1982.

HENDERSON, J. **Visual Identity Design**. Rockport Publishers, 2014.

HOLLOWAY, C. A.; NELSON, R. T. Job scheduling with due dates and overtime capacity. **Management Science**, 21,68-78, 1974.

HOLMBERG, K.; ROENNGVIST, M.; YUAN, D. An exact algorithm for the capacitated facility location problems with single sourcing. **European Journal of Operational Research**, 113:544–559, 1999.

HOLTZBLATT, K. **Rapid contextual design: a how-to guide to key techniques for user-centered design**. Morgan Kaufmann, 1998.

HVOLBY, H. H.; STEGER-JENSEN, K. Technical and industrial issues of Advanced Planning and Scheduling (APS) systems. **Computers in Industry**, 61 (9): 845-851, 2010.

KALYANMOY DEB. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II" **IEEE Transactions on Evolutionary Computation**, vol. 6, no. 2, pp. 182-197, April 2002.

KALYANMOY DEB, AMRIT PRATAP, SAMEER AGARWAL, T. MEYARIVAN. "A Study of Control Parameters for the NSGA-II Algorithm" **IEEE Transactions on Evolutionary Computation**, vol. 6, no. 2, pp. 182-197, April 2002.

KARAT, J. M. Cognitive walkthroughs for user interface evaluation. **Proceedings of the Third International Conference on Human-Computer Interaction**, 1990.

KAYID, A. **The role of Artificial Intelligence in future technology**. 2020.

KENNERLY, M.; NEELY, A. Enterprise resource planning: analysing the impact. **Integrated Manufacturing System**, 103–113, February, 2001.

KILGER, C.; SCHNEEWEISS, L. **Demand fulfilment and ATP**. In: STADTLER, H.; KILGER, C. (Eds.), *Supply Chain Management and Advanced Planning—Concepts, Models, Software and Case Studies*, Berlin, pp. 161–175, 2002.

KIM, J. U.; KIM Y. D. Simulated annealing and genetic algorithms for scheduling products with multi-level product structure. **Computers and Operations Research**, 23 (9): 857-868, 1996.

KOLISCH, R.; HESS, K. Efficient methods for scheduling make-to-order assemblies under resource, assembly area and part availability constraints. **International Journal of Production Research**, 38, 207-228, 2000.

KONAK, ABDULLAH; COIT, DAVID W.; SMITH, ALAN E. Multi-objective optimization using genetic algorithms: A tutorial. **Reliability Engineering & System Safety**, v. 91, n. 9, p. 992-1007, 2006.

LACERDA, D. P.; DRECH, A.; PROENÇA, A.; JÚNIOR, J. A. V. **Design Science Research: método de pesquisa para a engenharia de produção**. Gestão & Produção. vol.20 no.4. São Carlos 2013 Epub Nov 26, 2013.

LEE, Y. H.; JEONG, C. S.; MOON, C. Advanced planning and scheduling with outsourcing in manufacturing supply chain. **Computers and Industrial Engineering**, 43 (1-2): 351-374, 2002.

LIN, B. M. T. Scheduling in the two-machine flowshop with due date constraints. **International Journal of Production Economy**, 70, 117-123, 2001.

LIN, C. H.; HWANG, S. L.; WANG, E. M. Y. The mythical advanced planning systems in complex manufacturing environment. **IFAC Proceedings Volumes (IFAC-Papers Online)**, 2006.

LIN, C. H.; HWANG, S. L.; WANG, E. M. Y. A reappraisal on advanced planning and scheduling systems. **Industrial Management and Data Systems**, 107 (8): 1212-1226, 2007.

LIU, C.; WANG, J.; LEUNG, J. Y.T.; LI, K. Solving cell formation and task scheduling in cellular manufacturing system by discrete bacteria foraging algorithm. **International Journal of Production Research**, 54(3): 923-944, 2016.

MANSON, N. J. **Is operations research really research?** Orion, v.22, n.2, p.155-180. 2006.

MEYR, H.; ROSIC, H.; SEIPL, C.; WAGNER, M.; WETTERAUER, U. Architecture of selected APS. **Supply Chain Management and Advanced Planning**, 349–366, 2008.

MEYR, H.; WAGNER, M.; ROHDE, J. **Structure of advanced planning systems**. In: STADTLER, H.; KILGER, C. (eds) Supply chain management and advanced planning, pp 75–78. Springer, Berlin Heidelberg New York, 2000.

MONDAL, Arnab. **A Complete guide to Understand Classification in Machine Learning**. Gurgaon, Haryana, Índia. Analytics Vidhya. 2021. Disponível em: < <https://www.analyticsvidhya.com/blog/2021/09/a-complete-guide-to-understand-classification-in-machine-learning/> > Acesso em: 20 out. 2021.

MOON, C.; SEO, Y.; YUN, Y.; GEN, M. Adaptive genetic algorithm for advanced planning in manufacturing supply chain. **Journal of Intelligent Manufacturing**, 17 (4): 509-522, 2006.

MOON, C.; KIM, J. S.; GEN, M. Advanced planning and scheduling based on precedence and resource constraints for e-plant chains. **International Journal of Production Research**, 42 (15), 2941–2955, 2004.

MOSTAGHIM, S., BADER J. "A survey of multiobjective nature-inspired optimization algorithms" **Swarm and Evolutionary Computation**, vol. 1, pp. 1-16, 2011.

NAHMIAS, S. **Production and operations analysis**. Irvin, Homewood, 1997.

NEUMANN, K.; SCHWINDT, C.; TRAUTMANN, N. Advanced production scheduling for batch plants in process industries. **Or Spectrum**, 24(3): 251-279, 2002.

NIELSEN, J. Usability Engineering. **Morgan Kaufmann Publishers**. 1994.

PALMER, G. J. A simulated annealing approach to integrated production scheduling. **Journal of Intelligent Manufacturing**, 7, 163-176, 1996.

PETRONI, A. Critical factors of MRP implementation in small and medium-sized firms. **International Journal of Operations & Production Management**, 22 (3), 329–348, 2002.

PRESSMAN, R. S. **Engenharia de Software: Um Estudo de Caso**. McGraw-Hill Education. 2018.

QIN, A. K.; SUGANTHAN, P. N.; HANSEN, N.; et al. Differential evolution algorithm with strategy adaptation for global numerical optimization. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 2, p. 398-417, 2009.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**, 4ªEd. Berkeley, California, Estados Unidos. Pearson Education Limited, 2021.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, v. 3, n. 3, p. 210-229, 1959.

SANTOS, A. DOS. **Seleção do método de pesquisa: guia para pós-graduando em design e áreas afins**. 22ª ed. Curitiba, PR: Insight, 2018.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, v. 2, n. 3, p. 1-21, 2021.

SAUNDERS, C. **Research Methods for Business Students**. Prentice Hall. 2009.

SHAPIRO, Jonathan. Genetic algorithms in machine learning. In: **Advanced Course on Artificial Intelligence**. Springer, Berlin, Heidelberg, 1999. p. 146-168.

SHEN, W. M.; WANG, L. H.; HAO, Q. Agent-based distributed manufacturing process planning and scheduling: a state-of-the-art survey. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews**, 36 (4), 563–577, 2006.

SHIKI, S. B.; FRANZONI, F.; SILVA, S.; BRITO, G. C. **Ajuste de modelos simplificados de hidrogeradores usando séries temporais de vibração e algoritmos genéticos**. Foz do Iguaçu, Paraná, Brasil. Universidade Estadual do Oeste do Paraná. 2010.

SHNEIDERMAN, B. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**. Pearson. 2016.

STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, 11(4), 341-359. 1997.

SILVA, J. M.; FERREIRA, A. C.; COSTA, E.; et al. A survey of evolutionary algorithms for data mining and knowledge discovery. **Journal of Heuristics**, v. 24, n. 3, p. 343-380, 2018.

SILVER, E. A.; PYKE, D. F.; PETERSON, R. **Inventory management and production planning and scheduling**. Wiley, New York, 1998.

STADTLER, H.; KILGER, C. **Supply Chain Management and Advanced Planning – Concepts, Models Software and Case Studies**. 3rd ed. Berlin: Springer, 2005.

STADTLER, H. Supply chain management and advanced planning—basics, overview and challenges. **European Journal of Operational Research**, Volume 163, Issue 3, Pages 575-588, 2005.

SUM, C. C.; HILL, A. V. A new framework for manufacturing planning and control systems. **Decision Sciences**, 24 (4), 739–760, 1993.

SUNG, C. S.; MIN, J. I. Scheduling in a two-machine flowshop with batch processing machine(s) for earliness/tardiness measure under a common due date. **European Journal of Operational Research**, 131, 95-106, 2001.

SUNDARAM, R. M.; FU, S. S. Process planning and scheduling. **Computer and Industrial Engineering**, 15, 296-307, 1988.

TALABIS, MARK; MCPHEARSON, ROBERT; MIYAMOTO, I.; MARTIN, JASON. **Information Security Analytics: Finding Security Insights, Patterns and Anomalies in Big Data**. Waltham, Massachusetts, EUA. Syngress. 2014.

TAKAHASHI, R. H.C. **Otimização escalar e vetorial Volume 3: Otimização Vetorial**. Departamento de matemática. Universidade Federal de Minas Gerais. Minas Gerais. Brasil. 2007.

TASCA, J. E. et al. An approach for selecting a theoretical framework for the evaluation of training programs. **Journal of European Industrial Training**, 34(7), 631-655, 2010.

TECUCI, G. **Artificial intelligence**. Wiley Interdisciplinary Reviews: Computational Statistics. 4. 10. 2012.

VIEIRA J.; DESCHAMPS, F.; VALLE, P. D. **Advanced Planning and Scheduling (APS) Systems: A Systematic Literature Review**. In: ISTE 28TH INTERNATIONAL CONFERENCE ON TRANSDISCIPLINARY ENGINEERING, 28, 2021, Londres. EasyChair. 2021.

VIEIRA, J.; DE AGUIAR, A. P. G.; DESCHAMPS, F. **Application of the DSR – Design Science Research method for the development of an APS – Advanced Planning and Scheduling software**. In: I Latin American Congress of Applied Technologies, 1, Curitiba, 2021.

WAGNER M. Demand planning. In: Stadtler, H., Kilger, C. (eds) **Supply chain management and advanced planning**, pp 97–115. Springer, Berlin Heidelberg New York, 2000.

WANG, W.; SIAU, K. Artificial Intelligence, Machine Learning, Automation, Robotics, Future of Work and Future of Humanity: A Review and Research Agenda. **Journal of Database Management**. 30. 61-79. 2019.

WORTMANN, J. C. in: BITITCI, U. S.; CARRIE, A.S. (Eds.) **Evolution of ERP Systems Strategic Management of the Manufacturing Value Chain**. Kluwer Academic. Boston, MA, pp. 11–23, 1998.

WRIGHT, G; GOODWIN, P. **Forecasting with judgment**. Wiley, New York, 1998.

WU, C. M. **Genetic-Algorithm-for-Job-Shop-Scheduling-and-NSGA-II**, GitHub Page, Genetic Algorithm, 2018. Disponível em: <<https://github.com/wurmen/Genetic-Algorithmfor-Job-Shop-Scheduling-and-NSGA-II/blob/master/README.md>>. Acesso em: 10 nov. 2022.

WYLIE, L. **A vision for the next-generation MRP II**. Scenario, pp. 300–339, Gartner Group, April, 1990.

ZATTAR, I. C. **Análise da aplicação dos sistemas baseados no conceito de capacidade finita nos diversos níveis da administração da manufatura através de estudos de caso**. 136. Dissertação - Universidade Federal de Santa Catarina, Florianópolis, 2004.

ZHONG, R. Y.; LI, Z.; PANG, L.Y.; PAN, Y.; QU, T.; HUANG, G. Q. RFID-enabled real-time advanced planning and scheduling shell for production decision making. **International Journal of Computer Integrated Manufacturing**, 26(7): 649-662, 2013.