

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA**

CAMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA

BACHARELADO EM ENGENHARIA MECATRÔNICA

GUSTAVO BELMONTE CIOCCARI

**INTEGRAÇÃO DE UM ALGORITMO DE
PREVISÃO DE DEMANDA A UMA LINHA DE
MANUFATURA SIMULADA UTILIZANDO
TECNOLOGIAS DA INDÚSTRIA 4.0**

Florianópolis - SC

2021

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA
CATARINA**

CAMPUS FLORIANÓPOLIS

DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA

BACHARELADO EM ENGENHARIA MECATRÔNICA

GUSTAVO BELMONTE CIOCCARI

**INTEGRAÇÃO DE UM ALGORITMO DE
PREVISÃO DE DEMANDA A UMA LINHA DE
MANUFATURA SIMULADA UTILIZANDO
TECNOLOGIAS DA INDÚSTRIA 4.0**

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Bacharel em Engenharia Mecatrônica.

Orientador: Prof. Gregory Chagas da Costa Gomes

Florianópolis - SC

2021

Ficha de identificação da obra elaborada pelo autor.

Cioccari, Gustavo

**INTEGRAÇÃO DE UM ALGORITMO DE PREVISÃO DE DEMANDA
A UMA LINHA DE MANUFATURA SIMULADA UTILIZANDO TECNOLOGIAS DA
INDÚSTRIA 4.0 / Gustavo Cioccarri ; orientação de
Gregory Chagas da Costa Gomes. - Florianópolis, SC,
2021.**

51 p.

**Trabalho de Conclusão de Curso (TCC) - Instituto Federal
de Santa Catarina, Câmpus Florianópolis. Bacharelado
em Engenharia Mecatrônica. Departamento
Acadêmico de Metal Mecânica.**

Inclui Referências.

**1. Machine Learning. 2. Indústria 4.0. 3. Tecnologia.
4. Processos Produtivos. 5. Integração. I. Chagas
da Costa Gomes, Gregory. II. Instituto Federal de Santa
Catarina. Departamento Acadêmico de Metal Mecânica.
III. Título.**

**INTEGRAÇÃO DE UM ALGORITMO DE PREVISÃO DE DEMANDA A
UMA LINHA DE MANUFATURA SIMULADA UTILIZANDO
TECNOLOGIAS DA INDÚSTRIA 4.0**

GUSTAVO BELMONTE CIOCCARI

Este trabalho foi julgado adequado para obtenção do Título de Engenheiro em Mecatrônica e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 20 de abril, 2021.

Banca Examinadora:

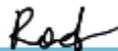


Gregory Chagas da Costa Gomes, Prof.



Mário de Noronha Neto (May 21, 2021 15:03 ADT)

Mário de Noronha Neto, Prof. Dr.



ROBERTO ALEXANDRE DIAS (May 20, 2021 09:43 ADT)

Roberto Alexandre Dias, Prof. Dr.

RESUMO

Com o advento da Indústria 4.0, a inclusão de novas tecnologias no processo produtivo é um diferencial muito grande na busca por um produto melhor e mais competitivo. Dentre essas novas tecnologias destaca-se o Machine Learning, que ainda não está muito difundida nas indústrias brasileiras, como mostram pesquisas recentes. Diante disso o objetivo do trabalho será estudar e utilizar algoritmos que indiquem a decisão mais adequada a se tomar na linha de produção, e integrá-los a essa através de protocolos de comunicação utilizados atualmente no ambiente industrial, implementando uma aplicação completa de ponta a ponta empregando tecnologias e conceitos chaves da Indústria 4.0. A fim de atingir esse objetivo estudou-se dois algoritmos de regressão linear para previsão de demanda, *Random Forest* e *Gradient Boost* obtendo resultados de 73% e 90% de assertividade, respectivamente, sendo esses satisfatórios para a integração proposta. Além disso, foi desenvolvida uma API para disponibilização dos dados obtidos pela implementação do algoritmo. A integração entre a API e o chão de fábrica se dá através de um middleware, com a função de API Gateway, onde é realizada a comunicação dos dados obtidos via OPC UA para a simulação do processo produtivo projetado. Também foi criado um dashboard para acompanhamento da produção em tempo de execução. Ao final foram realizados testes de comunicação entre as partes desenvolvidas para verificar o funcionamento da integração desejada e alcançou-se o objetivo proposto.

Palavras-chave: Machine Learning, Indústria 4.0, Tecnologia, Processos Produtivos, Integração.

ABSTRACT

With the advent of Industry 4.0, the inclusion of new technologies in the production process is a great differential looking forward to a better and more competitive product. In this context, it makes clear the need for the development of studies that demonstrate how Machine Learning can be used to achieve a more technological and well developed industrial environment. In addition to it, there is also the need to integrate this intelligence with the production environment in an automated way, to make it even more productive. Therefore, the goal of the work will be to study and use algorithms that indicate the most appropriate decision to be taken in the production line, and integrate them with it through communication protocols currently used in the industrial environment, implementing a complete application from end to end. using key technologies and concepts Industry 4.0. In order to achieve this goal, two linear regression algorithms for demand forecasting were studied, Random Forest and Gradient Boost. In addition, an API was developed to make available the data obtained by implementing the algorithms. By the development of a middleware, which function is being an API Gateway, the data obtained was sent via OPC UA to simulate the projected production process. A dashboard was also created to monitor production in run time. At the end, communication tests were carried out between the developed parts in the work to verify the functioning of the desired integration and the proposed objective was achieved.

Keywords: Machine Learning, Industry 4.0, Technology, Productive Processes, Integration.

1 INTRODUÇÃO	7
1.1 Definição do problema	9
1.2 Objetivos	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 Justificativa e Relevância	11
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 Fábricas Inteligentes	13
2.2 Inteligência Artificial	14
2.2.1 Machine Learning	15
2.3 Algoritmos de Machine Learning	15
2.3.1 Random Forest	16
2.3.2 Gradient Boost	17
2.5 OPC DA	19
2.6 OPC UA	20
3 PROPOSTA DE SOLUÇÃO	20
3.1 Modelo de previsão de demanda	22
3.1.1 Estrutura do código do modelo	24
3.2 Disponibilização dos dados de previsão via API Rest	27
3.3 API Gateway	30
3.4 Dashboard	34
3.5 CLP	36
3.6 Linha de manufatura simulada	37
4 RESULTADOS E DISCUSSÕES	39
4.1 Modelo de previsão de demanda	39
4.2 API REST	40
4.3 Comunicação API Gateway e CLP	42
4.4 Comunicação entre CLP e Dashboard	43
5 CONCLUSÃO	44
REFERÊNCIAS	46

1 INTRODUÇÃO

Desde o nascimento dos processos industriais de produção até hoje, diversas transformações ocorreram na cadeia de produção. Já presenciamos algumas mudanças de paradigma da produção nos últimos séculos, chamadas de revoluções industriais. Atualmente está ocorrendo uma nova mudança no paradigma de produção e consumo, muito incentivada pelo crescimento de potências econômicas antes pouco industrializadas, mas, principalmente, impulsionadas pela nossa nova relação com a internet em todos os níveis da sociedade e do nosso comportamento.

Desde o início do século XXI setores como o de logística, comércio, saúde e manufatura, de maneira geral, entre outros, se apropriaram de tecnologias ligadas à internet e a tecnologias da informação, dando início a algo que ainda não tinha nome, mas já começava a mudar as relações de consumo de produtos e serviços. Já estávamos presenciando os primeiros movimentos do que hoje pode ser chamado de 4ª revolução industrial.

Conceito que teve sua primeira aparição na feira de Hannover na Alemanha no ano de 2011 e que foi se consolidando até um conceito mais definido em 2013, a Indústria 4.0, nome dado à essa nova revolução industrial por um consórcio alemão entre governo e empresas deste país, tem como base a integração entre todos os elementos da cadeia produtiva através de alguns pilares que surgiram com o avanço da tecnologia como pode ser observado na Figura 1.

Em uma postagem em seu blog a empresa Siteware (2020) afirma que vivemos hoje a era da Indústria 4.0. Essa nova era de modelo industrial, além de otimizar as máquinas e processos produtivos do chão de fábrica, visa também uma integração com a demanda de mercado de forma mais precisa, utilizando computação para interpretar cálculos estatísticos com base nos dados disponíveis para auxiliar na tomada de decisão.

Diversos pesquisadores e empreendedores colocam a Inteligência Artificial em conjunto com a Ciência de Dados como um dos principais pontos da evolução dos processos produtivos que a Indústria 4.0 traz. Segundo Heuer (2020, tradução nossa) “executivos que adotarem essa importante evolução em uma nova era econômica não irão apenas sobreviver, mas também descobrir novas áreas de crescimento.”

Figura 1 - Pilares da Indústria 4.0



Fonte: Altus (2019)

Segundo Lakhani (2020, tradução nossa) essa importância se dá devido ao fato de que “Algoritmos de funções específicas ficam melhores com o passar do tempo à medida que obtêm mais e mais dados e ajustam os parâmetros”. Lakhani também afirma com relação a aplicação de Inteligência Artificial nas empresas que “Não importa se você faz hambúrgueres ou carros, a Indústria da Inteligência Artificial é a mesma em todas as empresas. É uma capacidade genérica que todas as organizações precisarão ter se quiser tirar vantagem do que está acontecendo.”

O presidente da Confederação Nacional da Indústria (CNI) afirma que

“As tecnologias digitais, quando são aplicadas de forma ampla na atividade industrial, produzem benefícios como aumento da eficiência operacional e redução de custos, flexibilização das linhas de produção e encurtamento dos prazos de lançamento de produtos, além de criação de produtos, serviços digitais e modelos de negócio. **A implementação das tecnologias habilitadoras da Indústria 4.0 é decisiva para a competitividade das empresas** e para a maior integração do país às cadeias globais de valor.” (ANDRADE, 2018, grifo nosso)

Nota-se que o uso de inteligência artificial integrado às plantas de produção mostra-se promissor por trazer diversos benefícios ao processo, dentre os quais podemos destacar:

- maior qualidade nos produtos;
- diminuição de custos;
- tomada de decisão mais assertiva;
- aumento da produtividade.

Tendo em vista o que foi dito pelos autores citados anteriormente, o trabalho será desenvolvido utilizando-se dos conceitos da indústria 4.0 e de inteligência artificial a fim de demonstrar o que poderia ser uma aplicação no cenário industrial brasileiro.

1.1 Definição do problema

Grande parte do processo de planejamento e controle da produção envolve a capacidade humana de processar informações e tomar decisões. Quanto mais complexo o negócio, mais demanda é gerada para quem atua nessa área. Apesar de todas as ferramentas existentes hoje que auxiliam nesse processo, a capacidade que um ser humano tem para essa função é limitada e quanto mais informações se tem do processo, mais complexa se torna a tomada de decisão (GASSER, 2014, tradução nossa).

Gasser (2020, tradução nossa) comenta que os modelos de tomada de decisão formulados por Tversky and Kahneman (1974) sugerem que seres humanos utilizam a informação do contexto para escolher entre alternativas, como levar mais em consideração potenciais de perda do que potenciais de ganho ou mesmo subestimar ou superestimar probabilidades.

Previsões de demanda desempenham um papel-chave em diversas áreas na gestão de organizações. A área responsável pela produção das fábricas utiliza-se dessas previsões no planejamento de modificações no nível da força de trabalho e no agendamento e priorização da produção (KRAJEWSKI & RITZMAN, 1999).

O processo de tomada de decisão no ambiente industrial hoje em dia na grande maioria das indústrias brasileiras é baseado em critérios objetivos porém com análises subjetivas. As fábricas que possuem formas de aquisição e controle

dos seus dados de produção e vendas conseguem tomar decisões melhores do que as que não possuem essas informações, de fato. Porém, esse processo pode ser largamente otimizado utilizando técnicas de inteligência artificial e machine learning para análises mais objetivas dos dados e integração dessas para com o ambiente fabril de produção.

A previsão de demanda utilizando métodos quantitativos pode ser feita através de vários modelos matemáticos. O emprego de cada modelo depende basicamente do comportamento da série temporal que se deseja analisar. Uma série temporal pode exibir até quatro características diferentes em seu comportamento: média, sazonalidade, ciclo e tendência (MAKRIDAKIS et al., 1998). Dessa forma, pode-se então aplicar algoritmos de machine learning para tornar o processo de análise computacional.

Além de criar modelos de machine learning é necessário integrá-los com o ambiente fabril a fim de automatizar e, assim, tornar mais eficiente o planejamento e controle da produção. Existem diversas formas de implementar essa integração de Inteligência Artificial nas indústrias e, então, criar um ambiente de produção inteligente (FUJIMAKI, 2020). Esse fato pode gerar confusão na hora de escolher as tecnologias corretas a serem implementadas.

Fujimaki (2020) em seu artigo para o site *Industry Today* também pontuou cinco desafios existentes hoje em boa parte das fábricas que precisam ser superados para obter-se um ambiente mais assertivo e conseqüentemente mais produtivo, sendo eles:

- Maquinário antigo e sistemas desconectados;
- incapacidade de acessar e aproveitar dados;
- demanda para resposta em tempo real;
- necessidade de times de tecnologia multidisciplinares;
- preconceito contra Inteligência Artificial.

Como foi apresentado, é uma dor das indústrias conseguir realizar a integração de machine learning ao ambiente de chão de fábrica, por isso busca-se com esse trabalho apresentar uma alternativa viável para essa implementação, utilizando tecnologias que já são comuns a esses ambientes para realizar a comunicação entre as partes. Além de ilustrar essa implementação utilizando ferramentas de simulação.

O que de fato será resolvido, então, com este trabalho é a seleção de tecnologias possíveis para realização da integração de um algoritmo de machine learning com o chão de fábrica, através de um modelo de previsão de demanda que, com base em dados históricos de consumo, conseguirá realizar a priorização da produção de forma automatizada.

1.2 Objetivos

1.2.1 Objetivo Geral

Utilizar recursos de machine learning em um algoritmo de auxílio à tomada de decisão orientada ao comportamento de mercado, integrando-o através de diferentes níveis e camadas de comunicação a um processo produtivo simulado.

1.2.2 Objetivos Específicos

- a) Buscar e manipular conjunto de dados de mercado (reais ou simulados) para implementação de um modelo de previsão de demanda;
- b) aplicar um algoritmo de machine learning para fazer uma análise de dados históricos e orientar tomada de decisão;
- c) analisar os resultados obtidos;
- d) disponibilizar dados obtidos em uma API;
- e) integrar dados obtidos com planta de produção simulada através de ferramentas de comunicação IIOT;
- f) exibir dados da produção do ambiente simulado em tempo de execução em dashboard web.

1.3 Justificativa e Relevância

De acordo com a pesquisa realizada pela Confederação Nacional da Indústria em 2018 sobre Investimentos em Indústria 4.0 houve um aumento significativo no número de indústrias brasileiras que utilizam pelo menos uma das tecnologias digitais consideradas pelo estudo, saltando de 63% para 73% (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2018).

Porém ainda de acordo com a pesquisa, a indústria brasileira não apresenta um grande passo na direção da digitalização, tendo 77,8% das indústrias com

plantas e sistemas de produção retrógrados, 20,6% com sistemas integrados utilizando mais de um dos pilares da Indústria 4.0 e apenas 1,6% conseguindo se manter atualizada ao mercado utilizando todos os pilares da Indústria 4.0, incluindo inteligência artificial.

O uso de aplicações de machine learning em sistemas de produção é uma tendência relativamente recente, no entanto, cada vez mais são os casos de sucesso de aplicações da mesma em processos de otimização, controle e previsão de falhas (SANTOS, 2018). Com todas as tecnologias disponíveis, os computadores devem ser treinados, preparados para cumprir tarefas específicas ao processar grandes quantidades de dados e reconhecer padrões nesses dados (SAS, 2020).

Esse tipo de técnica pode ser aplicado no planejamento e controle da produção de um ambiente industrial. Na condição de uma das áreas das empresas, o planejamento e controle da produção necessita de informações das áreas de marketing, suprimentos, engenharia, qualidade e manutenção e com isso torna-se necessária uma análise desses dados de forma conjunta. Guerrini, Belhot e Azzolini (2019) no livro Planejamento e Controle da Produção abordam como é feita a modelagem e implementação de um bom planejamento de produção. As atividades do PCP dizem respeito a identificar os sistemas de produção, prever vendas, planejar recursos, administrar estoques e programar atividades (GUERRINI; BELHOT; AZZOLINI Jr. 2019).

Os avanços nos conceitos de planejamento e controle da produção permitem refletir sobre o processo de aprendizado de máquina, motivando a busca por melhorias no modo de aplicar essas questões. A utilização dos simuladores ou das diversas ferramentas computacionais empregadas em ambientes reais de trabalho pode ajudar num melhor aprimoramento da produção (GIROTTI; MESQUITA, 2016).

Diante dos fatos apresentados é possível perceber que o desenvolvimento de pesquisas na área de machine learning e de tecnologias que permitam a integração desta com o ambiente de chão de fábrica e parte administrativa é importante para o auxílio na evolução dos processos produtivos e competitividade das indústrias nacionais.

2 FUNDAMENTAÇÃO TEÓRICA

Um dos desafios citados por Fujimaki (2020) para a implementação do conceito de indústria 4.0 nas plantas fabris é a falta de times multidisciplinares. Percebe-se com isso a necessidade de conhecimentos em diversas áreas da automação industrial, das tecnologias da informação e da ciência de dados para uma implementação básica deste paradigma. Neste sentido, este capítulo de fundamentação teórica busca reunir o mínimo de conceitos necessários para o entendimento da solução desenvolvida no trabalho e foram a base de estudos realizados pelo autor.

2.1 Fábricas Inteligentes

Radziwon et. al (2014) apresentam o conceito de Fábricas Inteligentes (em inglês, *Smart Factory*) como sendo:

“Uma solução de fabricação que fornece flexibilidade e processos de produção adaptativos que resolverão os problemas em uma instalação de produção com dinâmica e rápida mudança de condições de fronteira em um mundo de crescente complexidade” (RADZIWON et. al. 2014, tradução nossa).

Segundo Lucke et. al (2008) pode-se chegar ao modelo de fábricas inteligentes através de sistemas que trabalham em segundo plano, chamados sistemas *context-aware*. Ainda conceituam *context-aware* como sistemas que levam em consideração as informações provenientes do contexto do estado de produção da fábrica como por exemplo a posição de um produto na linha de produção. Esses sistemas cumprem suas tarefas com dados que podem ser originados tanto de dispositivos físicos, como sensores de posição ou do mundo virtual, como documentos eletrônicos ou modelos de simulação.

O planejamento e controle da produção de uma fábrica é influenciado por uma série de variáveis que determinam como deve ser feita a priorização da fabricação dos produtos. Bayazit (2004) em seu estudo *Use of AHP in decision-making for flexible manufacturing systems* listou variáveis que podem ser levadas em consideração no processo produtivo, sendo algumas delas:

- variedade de produtos;
- satisfação do cliente;

- tempo de preparação do maquinário;
- tempo de produção;
- produtividade;
- lucratividade
- disponibilidade de matéria prima.

Priore et. al. (2018) afirmam que a operação de sistemas de manufatura flexível baseados em *machine learning*, um dos braços da inteligência artificial, requer uma grande quantidade de dados para treinamento para modelamento do problema a ser resolvido e também dados de teste para que o modelo possa ser validado. Ainda expõem que esses dados podem ser obtidos do histórico de produção da linha ou até mesmo através de simulações de performance.

2.2 Inteligência Artificial

Apesar de estar bastante em foco nos dias de hoje como algo extremamente moderno, o conceito de inteligência artificial já era discutido desde meados do século XX. Em 1955 John McCarthy escreveu uma carta que continha uma proposta para a pesquisa de verão do Dartmouth College sobre inteligência artificial. Um trecho da carta comentava sobre o intuito da pesquisa:

“A tentativa será feita para descobrir como máquinas podem criar linguagem, formar abstrações e conceitos, resolver problemas restritos a humanos e até melhorar elas mesmas.” (MCCARTHY, 1956, tradução nossa)

McCarthy (2004) descreveu inteligência artificial como a ciência e engenharia de fazer máquinas e programas inteligentes com a tarefa de entender a inteligência humana, porém não limitando-se a métodos que um ser humano seria capaz de realizar.

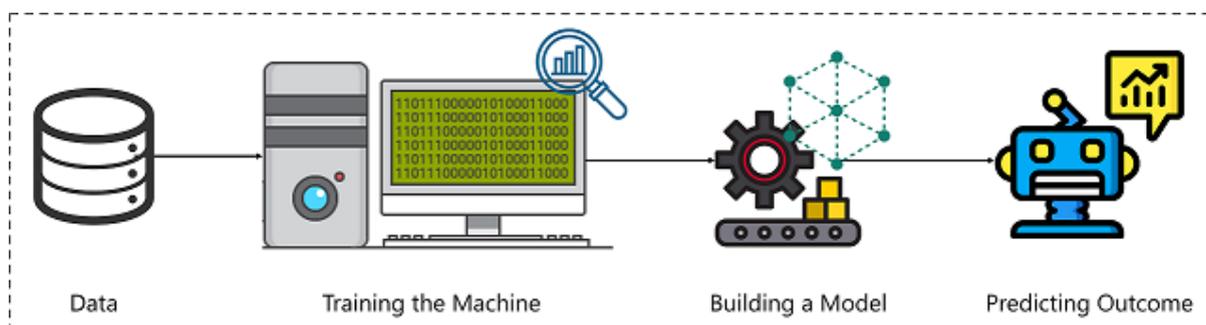
Inteligência artificial é o campo da ciência cujo objeto de estudo é “a síntese e análise de agentes computacionais que agem de forma inteligente” (POOLE e MACKWORTH, 2017, tradução nossa). Para Poole e Mackworth o principal objetivo científico da inteligência artificial é compreender os princípios que tornam possível comportamentos inteligentes em sistemas artificiais, através da análise de agentes, formulação e teste de hipóteses na construção de agentes inteligentes e por fim a experimentação desses agentes na realização de tarefas que requerem inteligência.

2.2.1 Machine Learning

O Expert.ai (2020) define o conceito de *machine learning* como sendo uma aplicação da inteligência artificial que permite aos sistemas a aprender e melhorar sua performance de acordo com a sua própria experiência executando as tarefas, tendo foco em acessar dados e usá-los para aprender por si próprio.

Em contraste com a inteligência artificial tradicional, a técnica de aprendizado de máquina não tem por objetivo imitar comportamentos inteligentes, mas sim usar as capacidades computacionais para complementar a inteligência humana, através da identificação de padrões em grandes conjuntos de dados, por exemplo. (SHALEV-SCHWARTZ e BEN-DAVID, 2014).

Figura 2 - Fluxo de funcionamento de *Machine Learning*



Fonte: Edureka (2020)

Mohri et al. (2018) afirmam que *machine learning* consiste em projetar algoritmos de predição eficientes e precisos, mas para que eles funcionem é necessário que a amostragem de dados seja compatível com o grau de complexidade do algoritmo.

2.3 Algoritmos de *Machine Learning*

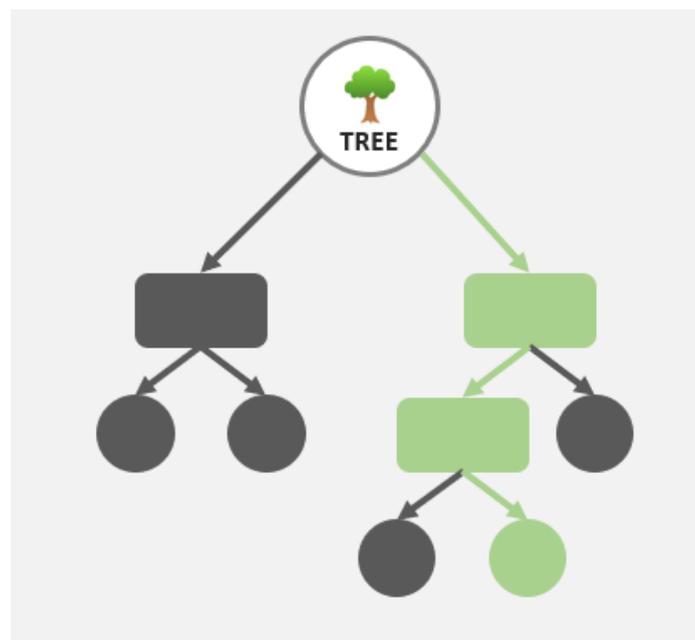
Os algoritmos de *machine learning* são divididos em basicamente duas fases, a fase de treinamento e a fase de teste. A fase de treinamento consiste em utilizar um conjunto de dados que servirão de base para o modelo gerado, é nessa fase que o algoritmo aprende as relações entre as variáveis presentes no conjunto de dados. Já na fase de teste o algoritmo irá comparar o modelo treinado com o conjunto de dados de teste e assim verificar a capacidade de predição do modelo gerado.

Existem diversos algoritmos já implementados e testados pela comunidade científica para diferentes tipos de aprendizado de máquina. Dentre eles estão o algoritmo de *Random Forest* e o algoritmo de *Gradient Boost*, que são utilizados no presente trabalho. Esses algoritmos foram escolhidos devido a sua ampla aplicabilidade nesse tipo de problema de previsão de demanda, encontrados nas pesquisas realizadas para estudo do tema.

2.3.1 *Random Forest*

O método random forest é baseado no algoritmo básico de árvores de decisão. O algoritmo de árvores de decisão é um método estatístico que utilizando um conjunto de dados cria uma representação do mesmo em forma de árvore. Cada nó é criado a partir das características dos dados fornecidos. Através de cálculos matemáticos são feitas escolhas por determinadas características como nó em uma posição definida da árvore.

Figura 3 - Esquema de árvore de decisão



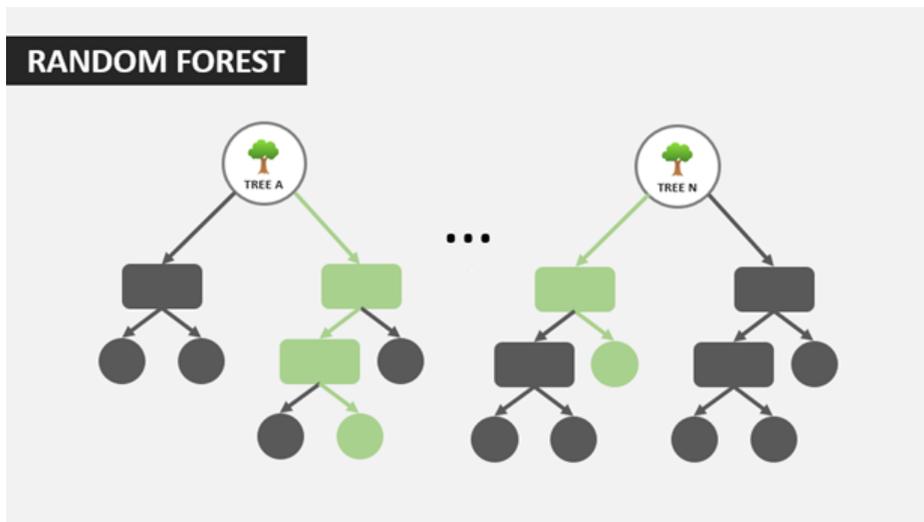
Fonte: PESSANHA (2019)

Dessa forma, como o nome sugere o algoritmo de *Random Forest* trabalha com diversas árvores de decisão da seguinte forma:

- 1) Seleciona aleatoriamente características do conjunto de dados fornecido;
- 2) Seleciona a característica mais adequada como o nó raiz;
- 3) Gera nós filhos;

- 4) Repete os passos anteriores até serem geradas a quantidade de árvores desejadas.

Figura 4 - Esquema de *random forest*



Fonte: PESSANHA (2019)

Esse funcionamento é baseado em *bagging* (*bootstrap aggregation*), ou seja, cada árvore de decisão gera um resultado e ao final esses resultados são comparados e o resultado obtido através de um processo determinístico de média é retornado como a resposta do algoritmo. Por esse motivo é um algoritmo que retorna uma previsão com certa precisão.

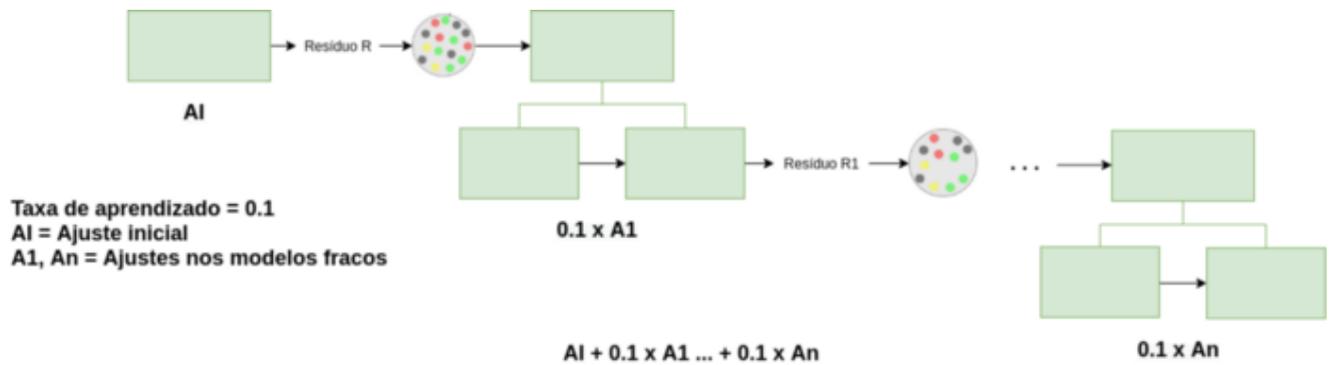
2.3.2 *Gradient Boost*

A técnica de *Gradient Boosting* consiste em gerar modelos de árvores de decisão considerados fracos e otimizá-los. Dessa forma, o funcionamento se dá pela minimização do erro através de uma função de perda a partir de cada modelo inicial gerado, portanto cada novo modelo gerado depende sempre do anterior.

A fim de determinar qual o impacto de cada modelo gerado no resultado final, em cada ajuste de modelo fraco ocorre uma multiplicação por um valor de taxa de aprendizagem. Quanto menor o valor dessa taxa, menor a contribuição que a árvore gera para o resultado final.

Uma representação gráfica dessa técnica pode ser visualizada na figura 5.

Figura 5 - Esquema de *Gradient Boosting*

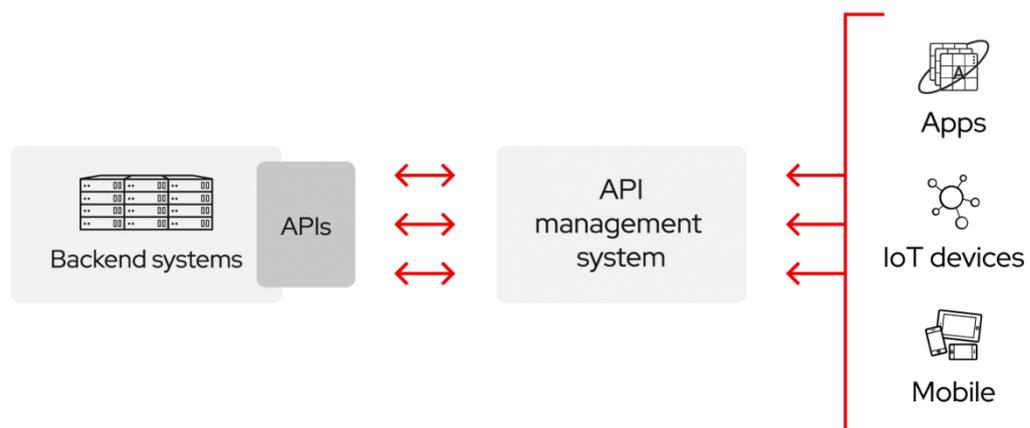


Fonte: SILVA (2020)

2.4 API REST

API é um acrônimo para *Application Programming Interface* em inglês que pode ser traduzido como Interface de Programação de Aplicação. Como o nome sugere, ela permite a comunicação entre diferentes sistemas de forma que um sistema que requisita acesso à API apenas utilize seus serviços sem necessariamente conhecer os detalhes da sua implementação. Essa implementação pode ser observada na figura 6.

Figura 6 - Representação gráfica da comunicação através de API.



Fonte: RedHat (2020)

Essa interface é geralmente implementada através de um modelo de requisição e resposta HTTP, contendo diferentes métodos para realização dos

serviços. Essa implementação na maior parte dos casos ocorre através de um padrão REST.

REST é um acrônimo para *Representational State Transfer* em inglês que pode ser traduzido como Transferência Representacional de Estado e foi introduzido em 2000 por Roy Fielding na sua tese de doutorado *Architectural Styles and the Design of Network-based Software Architectures*. Fielding (2000) propôs um guia com seis condições para seguir uma arquitetura REST. São elas:

1. Ter uma arquitetura cliente/servidor através de requisições HTTP;
2. Cada requisição deve ser não conectada uma com a outra, sem armazenar informações entre elas;
3. Armazenar dados em cache otimizando as interações cliente/servidor;
4. Ter um formato padronizado de transmissão de informações;
5. Ter um sistema em camadas hierárquicas que restringem o comportamento de cada componente para que ele não “veja” além da camada em que está interagindo;
6. Possibilitar código sob demanda (opcional), podendo enviar um código executável do servidor para o cliente quando solicitado.

2.5 OPC DA

OPC DA é a sigla para *Open Platform Communications - Data Access*. É um padrão de comunicação criado pela *OPC Foundation* que implementa grupos através do padrão cliente/servidor. Essa comunicação se dá através do itens *OPC*, dessa maneira o cliente escolhe quais itens deseja monitorar e assim estabelece a comunicação com o servidor.

Todo o acesso aos itens *OPC* é baseado em uma interface C/C++ utilizando-se da tecnologia de comunicação COM/DCOM da Microsoft através do sistema operacional Windows. Dessa forma, a padronização criada via OPC DA permite que seja estabelecida uma comunicação entre hardwares de diferentes fornecedores sem que seja necessária a instalação de diferentes softwares.

Por muitos anos o OPC DA foi uma solução muito utilizada na indústria para a integração de equipamentos de diferentes fabricantes operando com diferentes protocolos industriais de comunicação com sistemas de supervisão e gerenciamento de dados de produção.

2.6 OPC UA

Com o surgimento da internet e da necessidade de maior interoperabilidade entre plataformas, o OPC DA já não se mostrava uma solução viável. Mas, podemos dizer que ele evoluiu junto com essas necessidades, dando origem ao OPC UA.

OPC UA é a sigla para *Open Platform Communications - Unified Architecture* e como o nome sugere é um protocolo de comunicação industrial aberto. É utilizado para comunicação horizontal entre máquinas (modelo machine to machine) e comunicação vertical de máquinas para a nuvem. Segundo a *OPC Foundation* (2008), criadora do protocolo o projeto visou atingir os seguintes objetivos:

- Equivalência funcional: todas as especificações OPC clássicas são mapeadas para UA;
- Independência de plataformas: poder ser implementado desde microcontroladores até em infraestruturas baseadas em nuvem e também em sistemas operacionais diferentes;
- Segurança: encriptação, autenticação e auditoria;
- Extensão: possibilidade de adição de novas *features* sem afetar outras aplicações existentes;
- Modelagem compreensiva de informação: definição de informações complexas.

3 PROPOSTA DE SOLUÇÃO

Por se tratar de um trabalho de pesquisa envolvendo tecnologias novas ao autor, utilizou-se uma metodologia não linear no seu desenvolvimento. As etapas de revisão bibliográfica, planejamento e codificação das diferentes camadas de software desenvolvidas se revezaram durante o período da sua realização.

Tendo em vista o objetivo geral do trabalho de desenvolver uma integração de inteligência artificial a um ambiente de produção simulado, torna-se necessária a integração de diferentes camadas de software através de diferentes protocolos de comunicação. Com isso, foram avaliadas diferentes estratégias, arquiteturas e tecnologias para realizar a implementação da solução idealizada.

Por se tratar de um problema de engenharia, é possível encontrar diversas soluções para o mesmo problema. Neste trabalho foi necessário encontrar soluções para atender os seguintes pontos, definidos nos objetivos específicos:

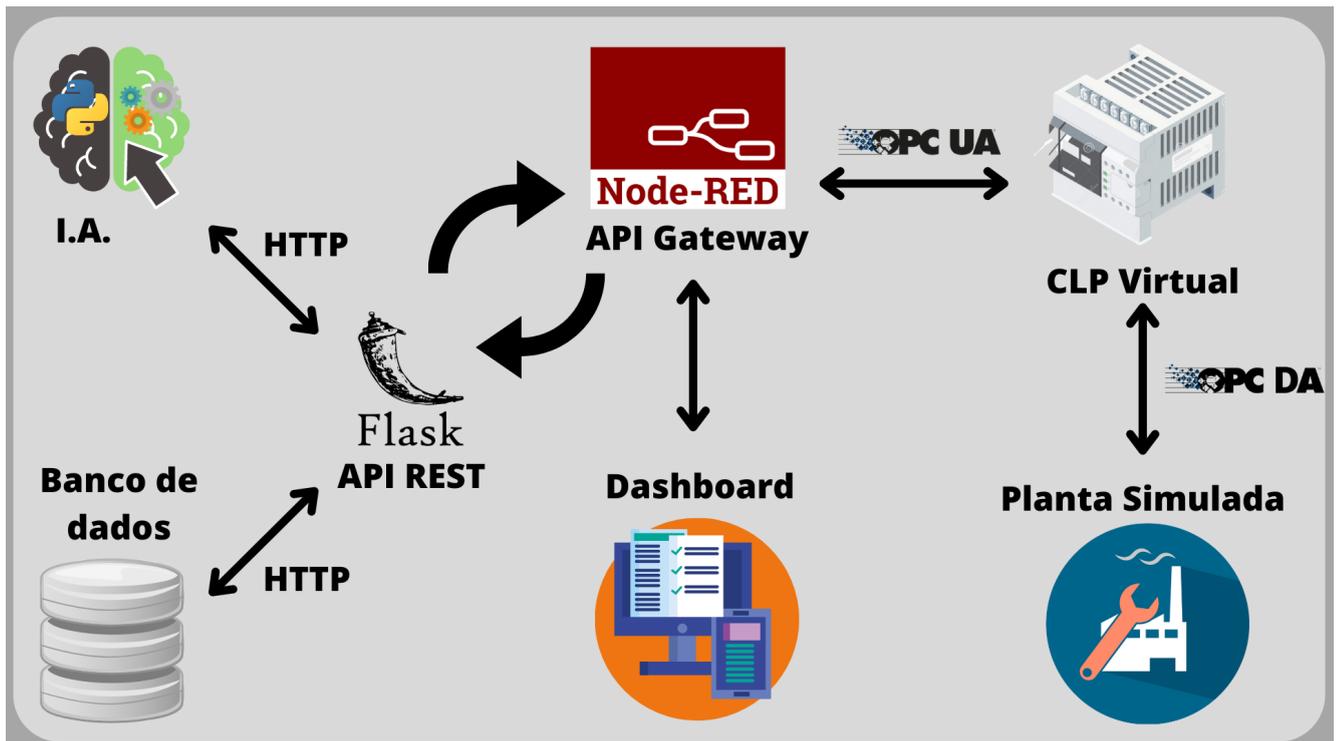
- Buscar e manipular conjunto de dados de mercado (reais ou simulados) para implementação de um modelo de previsão de demanda;
- aplicar um algoritmo de machine learning para fazer uma análise de dados históricos e orientar tomada de decisão;
- analisar os resultados obtidos;
- disponibilizar dados obtidos em uma API;
- integrar dados obtidos com planta de produção simulada através de ferramentas de comunicação IIot;
- exibir dados da produção em tempo de execução em dashboard web.

Cada objetivo gerou uma demanda de pesquisa e desenvolvimento, onde a solução desenvolvida envolve:

- a manipulação de um conjunto de dados históricos de demanda;
- o desenvolvimento de um modelo de previsão de demanda aplicando algoritmos de *Random Forest* e *Gradient Boost Regressor*;
- a criação de uma *API Rest* para disponibilização dos dados de previsão gerados;
- a implementação de um API gateway que comunica-se com um CLP virtual;
- a programação de um CLP virtual;
- a modelagem de um chão de fábrica em ambiente de simulação;
- a implementação de uma dashboard para visualização dos dados de produção;
- a integração de todas as camadas de software envolvidas.

A figura 7 apresenta a arquitetura geral do projeto, com o fluxo de comunicação entre as partes envolvidas no processo de integração da inteligência artificial com o processo de manufatura simulado.

Figura 7 - Arquitetura geral do projeto



Fonte: O autor (2021)

Com o intuito de facilitar o entendimento da arquitetura apresentada na figura 7, cada elemento presente no fluxo apresentado será discutido e apresentado sua forma de implementação em subseções deste capítulo, seguindo uma ordenação *top to bottom*.

3.1 Modelo de previsão de demanda

O modelo de previsão de demanda desenvolvido foi aplicado a um conjunto de dados disponibilizado na plataforma Kaggle¹ de forma pública. Esse conjunto contém dados de venda de diferentes lojas e produtos em um período de 4 anos.

Para o desenvolvimento deste trabalho foram utilizados dados de 4 produtos diferentes de uma única loja, totalizando 3599 linhas entre dados de treinamento e teste.

No desenvolvimento do modelo utilizou-se as ferramentas abaixo:

- Python;
- SciKit-Learn;

¹ Kaggle é uma plataforma com a maior comunidade de ciência de dados e que dispõe de recursos para ajudar quem atua na área. A plataforma permite aos usuários publicar e encontrar conjuntos de dados de naturezas diversas. O conjunto de dados utilizado pode ser encontrado em: <https://www.kaggle.com/c/demand-forecasting-kernels-only>

- Pandas.

O Python vem se estabelecendo como uma das mais populares linguagens para data science e machine learning. Devido a seu alto nível de interatividade natural e um ecossistema maduro de bibliotecas científicas e sua frequente escolha para desenvolvimento de algoritmos e análise de dados exploratória (DUBOIS, 2007; MILMANN e AVAIZIS, 2011).

A linguagem escolhida para trabalhar foi o Python devido a sua excelente aplicabilidade em códigos de inteligência artificial e também a ampla comunidade criada ao redor desse universo, contando com diversas bibliotecas, documentações e projetos práticos para tomar como base (RASCHKA e MIRJAILLI, 2019).

A biblioteca Scikit-Learn dá acesso ao usuário a uma grande variedade de algoritmos de machine learning, utilizando uma consistente interface orientada a tarefas, permitindo uma fácil comparação entre métodos para uma determinada aplicação. Por estar associada ao ecossistema do Python, ela pode ser facilmente integrada em aplicações além do escopo tradicional de análise de dados estatística (GRISEL et.al. 2011). Por esses motivos foi escolhida para ser utilizada no desenvolvimento do modelo de previsão. Neste trabalho foram utilizados os métodos de Random Forest Regressor e Gradient Boost Regressor, além do método *R-squared Score* para verificar a precisão do modelo gerado.

A biblioteca Pandas (MCKINNEY, 2010) foi utilizada pois facilita a manipulação de arquivos de diferentes tipos como por exemplo .csv, formato que foi disponibilizado o conjunto de dados utilizado para geração do modelo.

Para o desenvolvimento do código foi utilizada a plataforma Jupyter Notebook (PÉREZ e GRANGER, 2015), que permite a implementação de código em Python (além de outras linguagens) de forma facilitada, dividida em células. Essa estrutura auxilia na criação de um código organizado em partes, podendo executar e visualizar saídas de pedaços de código independentes uns dos outros. Essa possibilidade de visualizar a saída de cada célula em separado traz vantagens na utilização de bibliotecas de ciência de dados pois facilita o entendimento de cada etapa no processo de manipulação do conjunto de dados. Na figura 8 é possível visualizar um trecho de código que exemplifica o uso do Jupyter. O trecho executa um processo de carregamento dos conjuntos de dados e exibe na saída da célula os primeiros itens de cada conjunto carregado.

Figura 8 - Formato da tabela no banco de dados

Load Dataset

```
train = pd.read_csv("train.csv", parse_dates=['date'], index_col=['date'])
test = pd.read_csv("test.csv", parse_dates=['date'], index_col=['date'])

train = train[train.item <= 4]
test = test[test.item <= 4]

print(f"Shape: train={train.shape}, test={test.shape}")
print(train.head())
print(test.head())
```

```
Shape: train=(73040, 3), test=(3600, 3)
      store item sales
date
2013-01-01    1    1   13
2013-01-02    1    1   11
2013-01-03    1    1   14
2013-01-04    1    1   13
2013-01-05    1    1   10
      id store item
date
2018-01-01    0    1    1
2018-01-02    1    1    1
2018-01-03    2    1    1
2018-01-04    3    1    1
2018-01-05    4    1    1
```

Fonte: O autor (2021)

3.1.1 Estrutura do código do modelo

O código criação do modelo de previsão foi dividido em três grandes grupos, sendo eles a implementação e avaliação dos modelos utilizando Random Forest Regressor e Gradient Boost Regressor, comparação dos resultados através da métrica de *R-squared Score* e por fim a aplicação dos modelos para gerar a previsão dos próximos meses de vendas.

Para obter um resultado mais preciso, foram criadas *features* separando as vendas por dia, semana, mês, ano e quadrimestre além de dia da semana, semana do ano e verificação de fim de semana. Depois de aplicar essas *features* ao conjunto de dados fez-se uma separação entre dados de treino e de teste, separando o conjunto de dados numa proporção de 70% dos dados para treinar o modelo e 30% dos dados para testar o modelo.

Figura 9 - Criação e aplicação das *features* para treinamento do modelo

Model Evaluating Features

```
data_df['day'] = data_df['date'].dt.day
data_df['month'] = data_df['date'].dt.month
data_df['quarter'] = data_df['date'].dt.quarter
data_df['year'] = data_df['date'].dt.year
data_df['dayofweek'] = data_df['date'].dt.dayofweek
data_df['weekofyear'] = data_df['date'].dt.isocalendar().week
data_df['is_weekend'] = data_df['date'].dt.dayofweek // 5
```

```
data_df.drop(['date', 'id', 'sales'], axis=1, inplace=True)
```

```
x = data_df[data_df['train_or_test'] == 'train']
backtesting_test = data_df[data_df['train_or_test'] == 'test']
```

```
x.drop(['train_or_test'], axis=1, inplace=True)
backtesting_test.drop(['train_or_test'], axis=1, inplace=True)
```

```
y=pd.read_csv('train.csv')
y = y[y.item <= 4]
y=y.sales
```

```
x_train,x_test,y_train,y_test=model_selection.train_test_split(x,y,test_size=0.30,random_state=2)
```

```
ssc = StandardScaler()
x_train_ssc = ssc.fit_transform(x_train)
x_test_ssc = ssc.transform(x_test)
```

Fonte: Scikit Learn (2021)

Ao final, aplicou-se os algoritmos de regressão citados anteriormente e para avaliação do modelo utilizou-se o indicador *R-squared Score*, que informa o percentual da variância na variável independente que valores independentes explicam de forma coletiva.

Figura 10 - Trecho que implementa a predição de vendas utilizando *Gradient Boost Regressor*

Gradient Boosting Regressor

```
gbr=GradientBoostingRegressor(max_depth=4)
gbr

GradientBoostingRegressor(max_depth=4)

# %%time
gbr.fit(x_train,y_train)

GradientBoostingRegressor(alpha=0.92, max_depth=4, n_estimators=200)

# %%time
y_pred_gbr=gbr.predict(x_test)
y_pred_gbr

array([14.61352636, 23.5646938 , 12.98447195, ..., 32.45898971,
       22.14382784, 36.55282021])
```

Fonte: Scikit Learn (2021)

O cálculo do indicador é implementado pela biblioteca SciKit Learn utilizada e sua fórmula pode ser observada na equação 1.

Equação 1 - Fórmula do indicador *R-squared Score*

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Fonte: Scikit Learn (2021)

R-squared Score mede a força da relação entre o modelo gerado e a variável dependente em uma escala de 0 a 100% (FROST, 2018).

Após a construção dos modelos, foram implementados os métodos de previsão da biblioteca utilizada, obtendo como resultado os dados de vendas para os três meses seguintes à última data presente no conjunto de dados.

Não foi implementado nenhum método de aprimoramento dos parâmetros para os regressores pois o foco do trabalho reside na integração entre as partes e em ilustrar como poderia ser feita essa aplicação com as diferentes ferramentas apresentadas em um ambiente industrial.

3.2 Disponibilização dos dados de previsão via API Rest

O sistema de controle de produção do ambiente de produção simulado deve conseguir tomar decisões baseadas nos dados de demanda futura previstos pelo modelo construído. Tratando-se de um sistema distribuído, onde o sistema de produção e o sistema de previsão não se encontram em uma mesma rede local, é necessário disponibilizar os dados de provisão do modelo desenvolvido de maneira segura e robusta - prioritariamente, seguindo os padrões que vêm sendo adotados na indústria 4.0. Dessa forma, optou-se por salvar os resultados em um banco de dados utilizando um script simples desenvolvido em Python e acessá-los através de uma API REST. O script desenvolvido agrupa as previsões para cada item por semana antes de salvar no banco de dados, dessa forma a produção poderia ser planejada semanalmente. O formato da tabela no banco de dados pode ser visualizado na figura 10.

Figura 10 - Formato da tabela no banco de dados

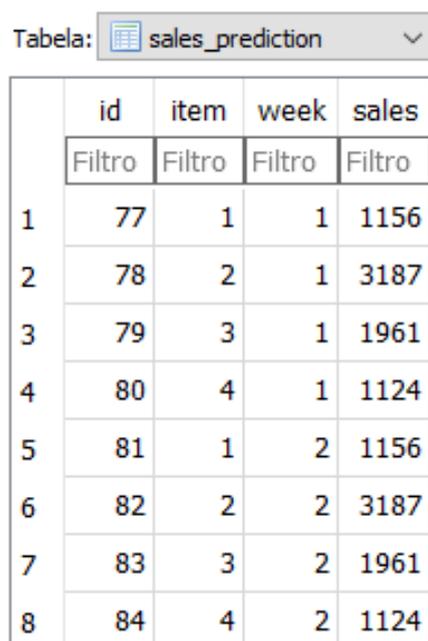


Tabela: sales_prediction

	id	item	week	sales
	Filtro	Filtro	Filtro	Filtro
1	77	1	1	1156
2	78	2	1	3187
3	79	3	1	1961
4	80	4	1	1124
5	81	1	2	1156
6	82	2	2	3187
7	83	3	2	1961
8	84	4	2	1124

Fonte: O autor (2021)

A vantagem de adotar essa abordagem é que os dados podem ser acessados por diferentes clientes, de acordo com a necessidade e sem que seja necessária uma implementação específica desses clientes, já que “uma API permite que sua solução ou serviço se comunique com outros produtos e serviços sem precisar saber como eles foram implementados” (REDHAT, 2020). Considerando o

fato apresentado e da necessidade de integração do modelo de previsão gerado com diferentes níveis de um processo produtivo, avaliou-se que a decisão por desenvolver uma API seria a mais adequada.

A API foi desenvolvida utilizando Flask (RONACHER, 2010), um microframework da linguagem Python para desenvolvimento web, utilizando métodos HTTP. O Flask foi escolhido pois se trata de uma ferramenta simples com arquitetura minimalista que permite uma implementação rápida sem a necessidade de servidores externos para rodar o serviço, além da sua facilidade para a integração com banco de dados, sem depender necessariamente de uma camada adicional para isso.

O serviço criado dispõe de rotas para acesso a dados de demanda prevista para 12 semanas seguintes à última presente no conjunto de dados utilizado para cada um dos 4 itens avaliados. Sendo assim, o usuário faz a requisição para a API informando como parâmetro a semana que ele precisa que sejam retornados os dados. O formato da requisição e da resposta podem ser visualizados na tabela 1.

Tabela 1 - Exemplo de requisição e resposta da API para a previsão da semana 1

Método HTTP:	GET
Endpoint:	/sales/week/1
Status da resposta:	200
Formato da resposta:	<pre>{ "data": [{ "item": 1, "sales": 1156, "week": 1 }, { "item": 2, "sales": 3187, "week": 1 }, { "item": 3, "sales": 1961, "week": 1 }, { "item": 4, "sales": 1124, "week": 1 }] }</pre>

Fonte: O autor (2021)

3.3 API Gateway

A maioria dos dispositivos de campo, como um CLP, possui interfaces de comunicação para protocolos deste mesmo nível, sendo raros os CLPs com suporte ao protocolo HTTP, que possibilitaria a comunicação direta entre ele e a API desenvolvida.

Uma proposta de solução para esse problema seria a implementação de um sistema SCADA que realizaria a comunicação com a API e o CLP ao mesmo tempo. Tentou-se implementar essa arquitetura exposta, porém alguns fatores levaram a desistência no meio do processo, sendo eles:

- solução dependente da plataforma Windows e pouco flexível;
- falta de familiaridade do autor com a linguagem de programação utilizada na criação do script de requisição para a API (Visual Basic).

Tornou-se necessário, portanto, a busca por uma nova solução mais moderna que satisfizesse os requisitos necessários. Diante disso, foi realizado o desenvolvimento de um middleware com a função de API Gateway, cujo objetivo é permitir a dispositivos que não se comunicam via HTTP, como é o caso do CLP, o acesso à API. Dessa forma, diante do que foi desenvolvido, o gateway faz a requisição das quantidades que devem ser produzidas de cada item na semana especificada utilizando uma rota da API e encaminha os dados retornados através do protocolo OPC UA para o CLP simulado, que tem suporte a esse protocolo de comunicação. Sendo assim, ele é responsável por informar quantos itens devem ser produzidos de cada tipo ao CLP.

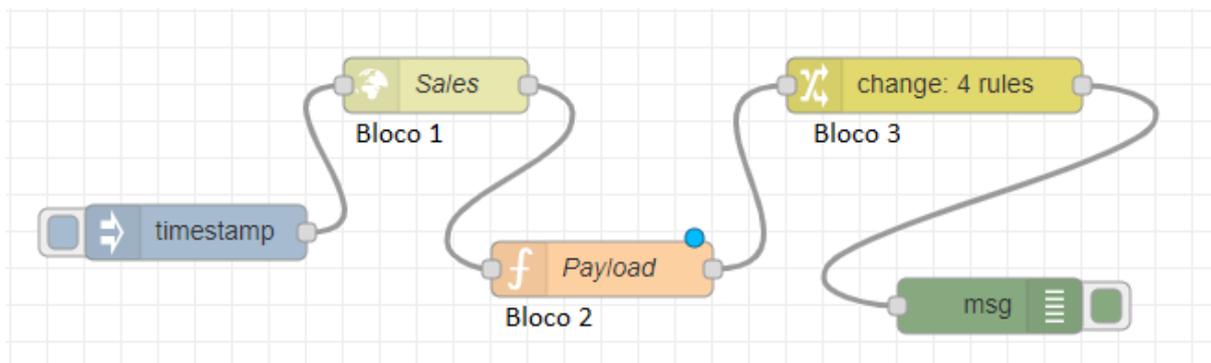
A programação do API gateway poderia ter sido feita em diversas linguagens de programação com bibliotecas para comunicação HTTP e OPC UA. Todavia, tendo em vista que um dos objetivos do trabalho é a implementação de um dashboard, foi decidido utilizar um software/plataforma que conseguisse abranger as duas funções de maneira mais simplificada, tanto a função de gateway, quanto de servidor de uma interface web para o dashboard. A ferramenta escolhida para a realização do proposto foi o Node Red que, apesar da menor robustez, é uma ferramenta mais abrangente e de simples prototipagem. O Node Red é uma ferramenta desenvolvida pela IBM em 2013 e que se tornou um projeto de código aberto para a *JS Foundation*. É uma plataforma que torna o acesso aos conceitos

de práticas de Internet das Coisas mais acessível ao público geral por ser baseada numa programação através de fluxos, em que cada “*node*” é uma “caixa preta” que recebe, processa e repassa dados, segundo a própria documentação disponibilizada no site da ferramenta (NODE-RED, 2013). Dessa forma é possível conectar hardwares, APIs e outros serviços em um único ambiente de programação.

Por ser uma plataforma de código aberto, a comunidade pode contribuir com as mais diversas bibliotecas para diferentes tipos de aplicações utilizando tecnologias de internet das coisas. Tal recurso possibilitou a utilização de uma biblioteca previamente desenvolvida e testada pela comunidade, que implementa blocos para comunicação via OPC UA, fornecendo então métodos como injetar, ler e escrever valores contidos em *nodes* do servidor OPC UA estabelecido no CLP virtual utilizado.

Na figura 11 é possível visualizar um exemplo do fluxo de requisição dos dados de previsão para a API desenvolvido no Node Red.

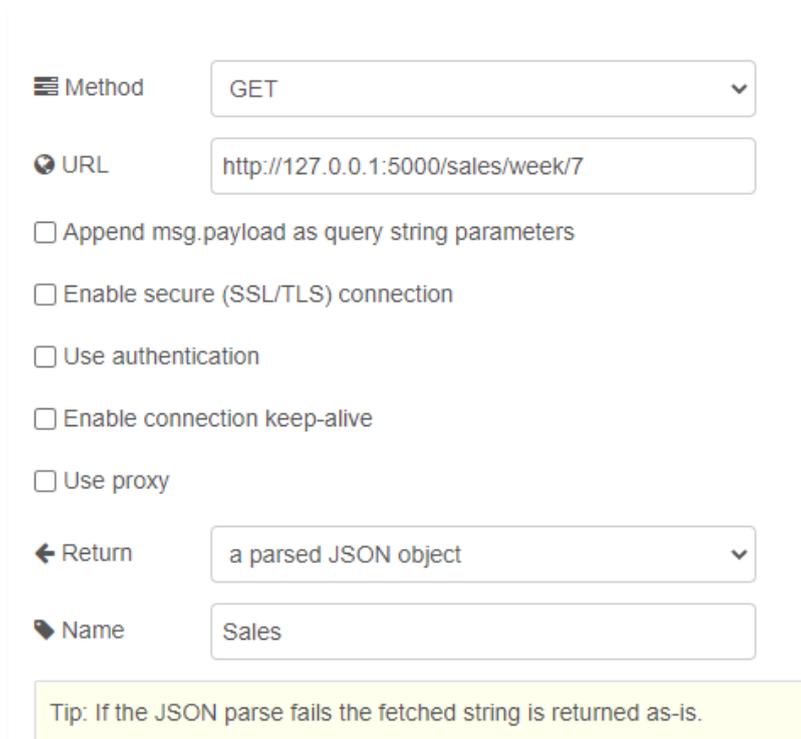
Figura 11 - Fluxo de requisição para API



Fonte: O autor (2021)

Na figura 12 pode-se verificar a configuração do bloco 1, responsável por realizar a requisição à API.

Figura 12 - Exemplo de configuração do bloco de requisição à API



Method: GET

URL: http://127.0.0.1:5000/sales/week/7

Append msg.payload as query string parameters

Enable secure (SSL/TLS) connection

Use authentication

Enable connection keep-alive

Use proxy

Return: a parsed JSON object

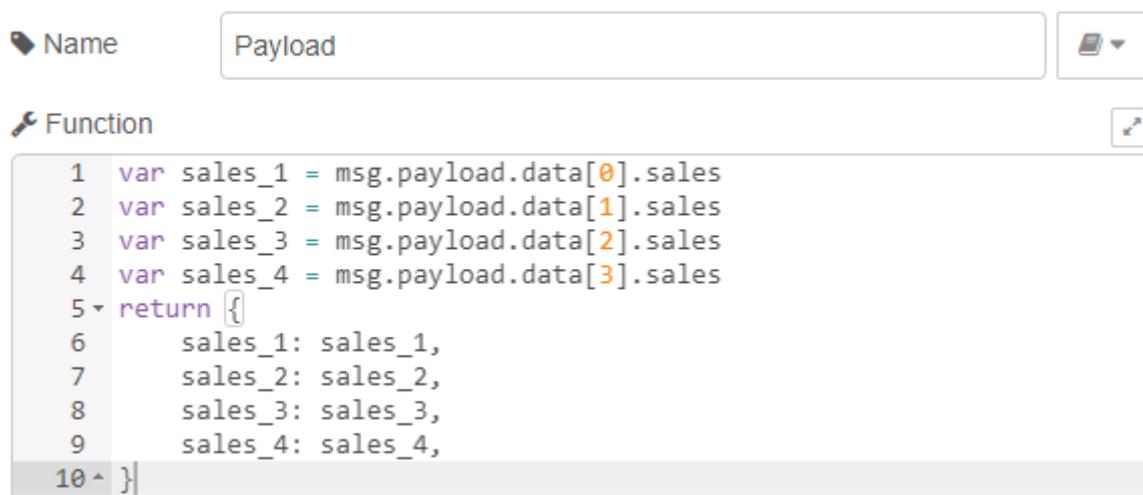
Name: Sales

Tip: If the JSON parse fails the fetched string is returned as-is.

Fonte: O autor (2021)

A API retorna um *payload* no formato JSON e no bloco 2 da figura 11 é realizado um *parsing* desse objeto, como mostra a figura 13.

Figura 13 - *Parsing* do objeto JSON retornado da API



Name: Payload

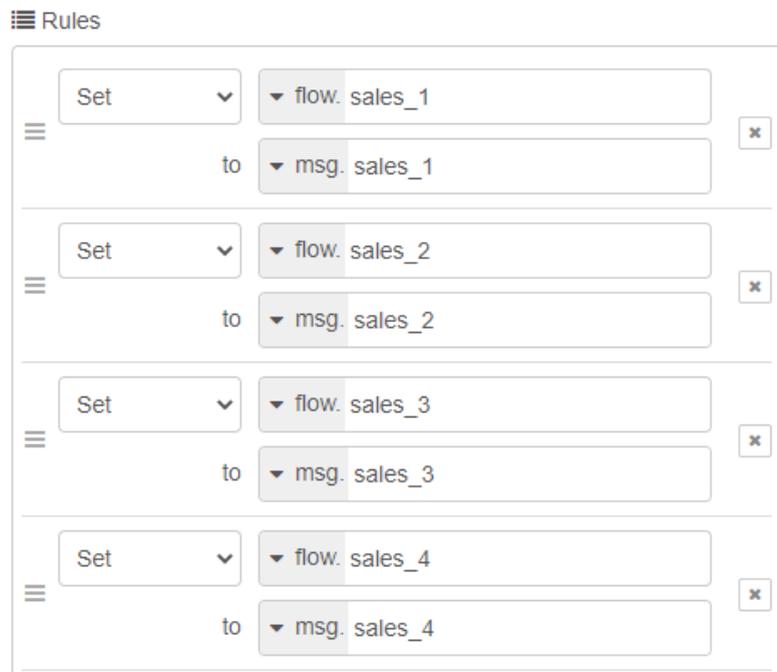
Function

```
1 var sales_1 = msg.payload.data[0].sales
2 var sales_2 = msg.payload.data[1].sales
3 var sales_3 = msg.payload.data[2].sales
4 var sales_4 = msg.payload.data[3].sales
5 return {
6   sales_1: sales_1,
7   sales_2: sales_2,
8   sales_3: sales_3,
9   sales_4: sales_4,
10 }
```

Fonte: O autor (2021)

Depois de efetuar essa operação, há um bloco de switch, identificado como bloco 3 na figura 11 que é responsável por salvar a demanda prevista de cada item para a semana requisitada em suas respectivas variáveis, como pode ser observado na figura 14.

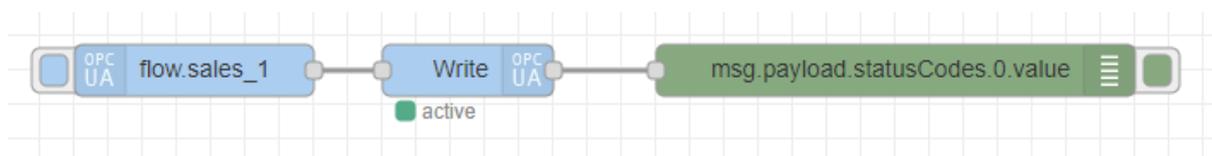
Figura 14 - Bloco responsável por salvar as variáveis



Fonte: O autor (2021)

Essas variáveis pertencem ao fluxo geral da aplicação e são utilizadas pelo fluxo de escrita OPC UA, informando ao CLP quantas peças devem ser produzidas de cada item. Esse fluxo de escrita no servidor OPC UA pode ser observado na figura 15.

Figura 15 - Fluxo de escrita no *node* OPC UA



Fonte: O autor (2021)

As figuras 16 e 17 mostram a configuração de conexão do servidor OPC UA e a configuração do bloco de entrada de valores no *node* OPC UA desejado, respectivamente.

Figura 16 - Configuração bloco de entrada de valores no *node* OPC UA

The screenshot shows the configuration interface for an OPC UA node. It features three tabs: 'Settings' (selected), 'Security', and 'Strategy'. Under the 'Settings' tab, there are several configuration options: 'Discovery' (with a search icon), 'Endpoint' (with a search icon and the value 'opc.tcp://DESKTOP-PP8R0J0:4840'), 'Endpoint Must Exist' (checkbox), 'Auto Select Endpoint' (checkbox), and 'Keep Session Alive' (checkbox, checked). Below these is a 'Name' field containing 'CODESYS SERVER'. At the bottom, there is a 'Show Errors' checkbox.

Fonte: O autor (2021)

Figura 17 - Configuração bloco de entrada de valores no *node* OPC UA

The screenshot shows the configuration interface for an OPC UA node, specifically the 'OPC UA Nodes' tab. It features two tabs: 'Settings' and 'OPC UA Nodes' (selected). Under the 'OPC UA Nodes' tab, there are two main configuration options: 'Type' (dropdown menu, value 'write') and 'Payload' (dropdown menu, value 'flow. sales_1'). Below this, there is a list of nodes. The first node is 'SALES' with a path 'ns=4;s=|var|CODESYS Control Wir' and a data type 'Int16'. There are search and delete icons next to the node list.

Fonte: O autor (2021)

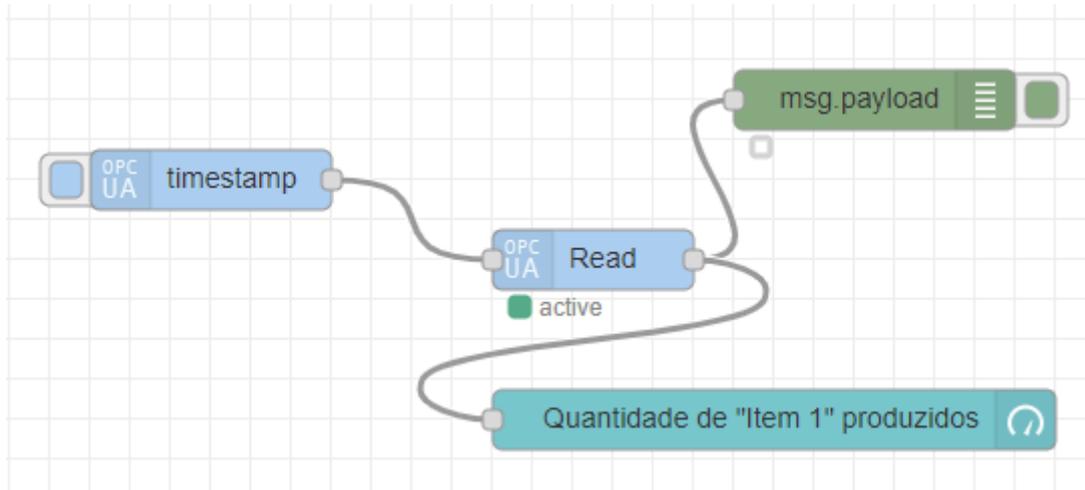
3.4 Dashboard

Como último objetivo do trabalho está o desenvolvimento de um dashboard web para monitoramento da produção.

Foi desenvolvido um servidor que responsabiliza-se por se comunicar com o CLP via OPC UA, e receber em tempo de execução quantos itens de cada tipo já foram produzidos, disponibilizando essas informações da produção simulada em um

dashboard via interface web. O fluxo de leitura no *node red* utilizando o OPC UA é demonstrado na figura 18. Esse fluxo é repetido para cada item que se deseja ler.

Figura 18 - Fluxo de leitura do *node* OPC UA



Fonte: O autor (2021)

O *node timestamp* é responsável por fazer a requisição periódica ao *node Read* (todos da biblioteca OPC UA), que se comunicam com o CLP e leem os valores de produção de cada item.

Figura 19 - Exemplo de configuração de Gauge

A interface de configuração de um Gauge mostra os seguintes campos:

- Group:** [TCC] TCC
- Size:** 6 x 4
- Type:** Donut
- Label:** Quantidade de "Item 1" produzidos
- Value format:** {{value}}
- Units:** units
- Range:** min 0, max 10
- Colour gradient:** Uma barra com gradientes de vermelho, amarelo e verde.
- Sectors:** 0 ... optional ... optional ... 10
- Name:** Campo de texto vazio.

Fonte: O autor (2021)

O *node gauge* está ligado na saída do *node read* e é o responsável por criar uma página web com um indicador gráfico que exibe o valor lido. Novos *nodes* desse tipo que forem adicionados ao fluxo irão adicionar novos elementos na mesma página web ou em uma diferente, de acordo com as configurações de *Group*. As possíveis configurações do *node gauge* são apresentadas na figura 19.

3.5 CLP

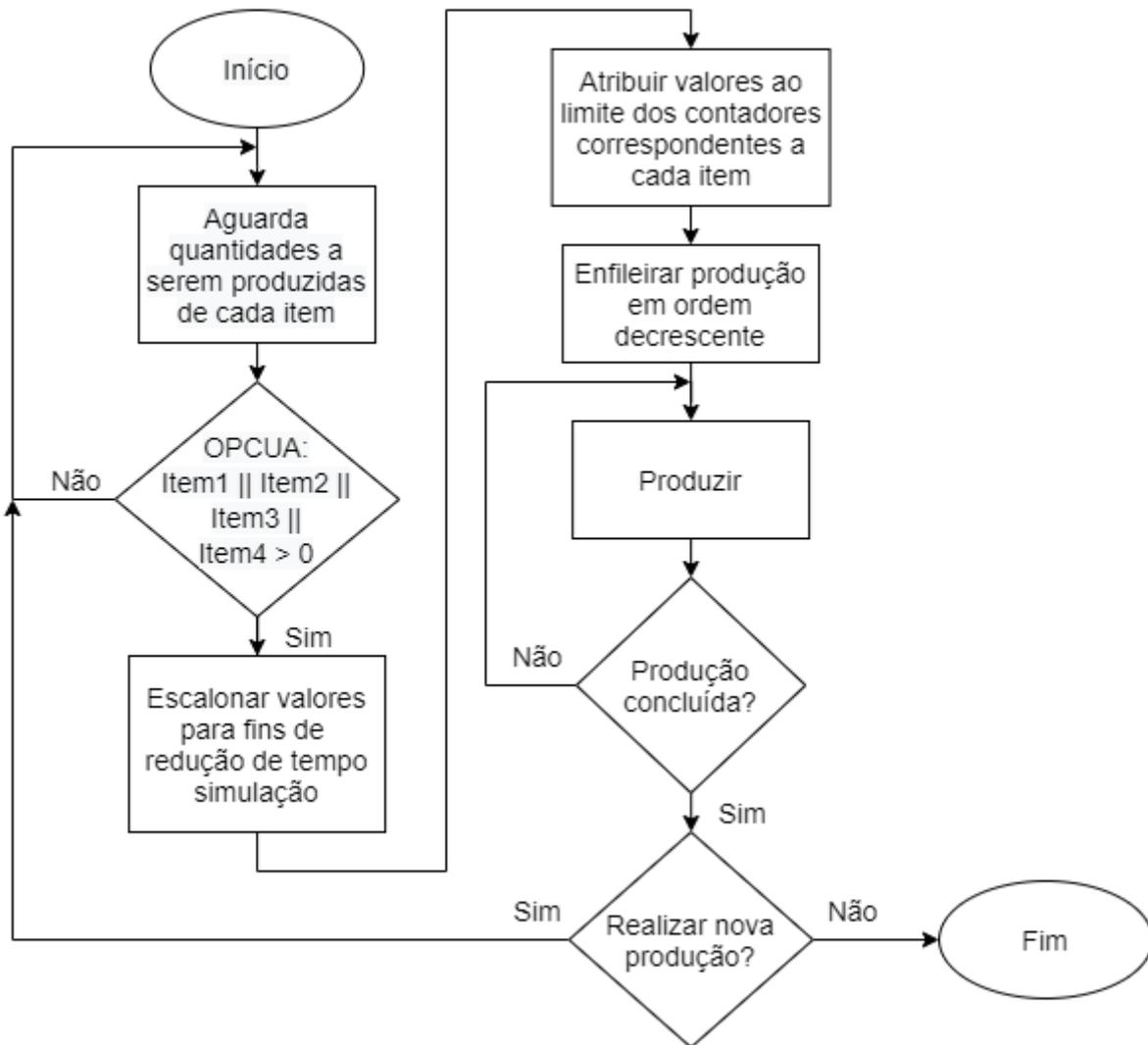
O CoDeSys foi o software escolhido para emular o CLP responsável por receber, processar e enviar sinais a fim de comandar a planta desenvolvida. O software segue a norma IEC 61131-3 para programação de CLPs e dentre as linguagens disponibilizadas pelo programa a escolhida para o desenvolvimento da lógica de atuação entre sensores e atuadores do processo produtivo simulado desenvolvido foi a linguagem Ladder.

O programa traz consigo um gateway para estabelecer uma comunicação com outros softwares através de algum dos protocolos de comunicação que estão disponíveis para utilização. Dessa forma o servidor fica alocado de forma local no computador do usuário.

Como demonstrado na figura 7, que ilustra as camadas da arquitetura do projeto, o gateway do CoDeSys estabelece um servidor OPC UA que comunica-se com o API Gateway e dashboard desenvolvidos no Node-Red os quais foram explicados na seção 3.4 e estabelece um servidor OPC DA utilizado na comunicação com a célula de manufatura simulada desenvolvida.

A lógica simplificada implementada no CLP virtual para validação do sistema de manufatura idealizado está ilustrada na figura 20.

Figura 20 - Lógica do CLP simplificada



Fonte: O autor (2021)

3.6 Linha de manufatura simulada

Para Harrell (2000), a simulação de sistemas de manufatura é uma importante ferramenta de planejamento gerencial que traz grandes contribuições, tais como: análise do impacto na variação do tamanho do lote de fabricação, controle da produção, planejamento da cadeia de suprimentos, programação da produção, controle da produção em tempo real, avaliação do impacto da introdução de um novo produto, previsão de gastos, criação de cenários e avaliação da eficácia dos planos de contingência.

Neste trabalho a simulação possui um papel fundamental de ilustrar a implementação de um sistema de inteligência artificial de previsão de demanda que tem por objetivo tornar mais eficiente o planejamento e controle de um processo

produtivo, utilizando tecnologias plenamente viáveis de serem integradas em um ambiente industrial.

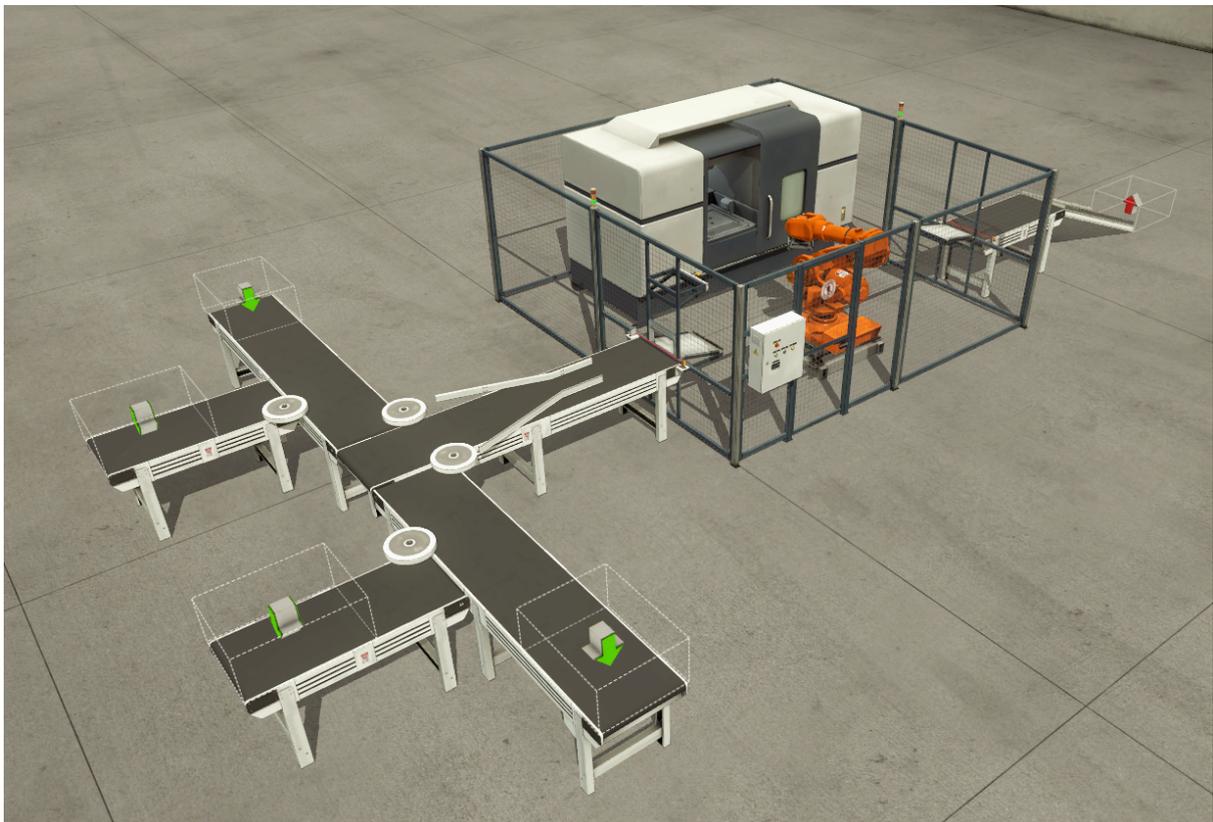
Com o intuito de tornar visível a integração entre cada parte da arquitetura, escolheu-se o software Factory IO para simular uma linha de manufatura simplificada para o desenvolvimento do trabalho.

O Factory IO é um software de simulação de plantas fabris e tem um intuito educacional de promover treinamentos de técnicas de automação. Ele permite uma fácil customização das plantas possibilitando a inserção de diversos dos principais maquinários, componentes e sensores de uma planta fabril.

Além disso, o programa permite uma fácil integração com controladores e CLPs (físicos ou virtuais) através de diversos protocolos de comunicação, como o ModBus e o OPC DA.

Para o desenvolvimento deste trabalho optou-se pela criação de uma linha de manufatura bastante simplificada para simular a integração que pode ser observada na figura 21.

Figura 21 - Linha de manufatura simulada



Fonte: O autor (2021)

Conforme citado, o Factory IO permite a utilização de protocolo OPC DA para comunicação de dados. Essa possibilidade e a sua fácil implementação foram fatores determinantes para a escolha desse software, pois o CLP virtual do CoDeSys implementa um servidor OPC DA, tornando assim possível a comunicação entre a programação do CLP com os atuadores e sensores da planta.

4 RESULTADOS E DISCUSSÕES

Tendo em vista os objetivos específicos estipulados na seção 1.2.2 os resultados deste trabalho dependem de 4 fatores, sendo eles:

- Assertividade do modelo de machine learning de previsão de demanda desenvolvido;
- sucesso na disponibilização dos dados obtidos através da API REST desenvolvida;
- sucesso na escrita dos valores dos dados retornados da API REST utilizando o API Gateway no CLP virtual via OPC UA;
- sucesso na leitura dos dados de produção disponíveis no CLP virtual em tempo de execução via OPC UA para disponibilização no dashboard.

Dessa forma, cada fator será abordado separadamente nesta seção.

4.1 Modelo de previsão de demanda

Conforme citado na seção 3.1 o indicador utilizado para avaliação dos modelos gerados foi o *R-squared Score*. Apesar de não terem sido aplicadas técnicas de aprimoramento de parâmetros, obteve-se valores considerados satisfatórios do indicador utilizado, sendo um *R-squared Score* de 73% para o algoritmo de *Random Forest Regressor* e de 90% para o algoritmo de *Gradient Boost Regressor*. A implementação, resultados e comparação deste indicador para cada algoritmo pode ser observada nas figuras 22, 23 e 24.

Figura 22 - Cálculo *R-Squared* para o *Random Forest Regressor*

Random Forest Regressor

```
y_pred_rf=rf.predict(x_test)
s2=r2_score(y_test,y_pred_rf)
print(s2)
```

0.7297360310048631

Fonte: O autor (2021)

Figura 23 - Cálculo *R-Squared* para o *Gradient Boost Regressor*

Gradient Boosting Regressor

```
s1=r2_score(y_test,y_pred_gbr)
print(s1)
```

0.8993032725302225

Fonte: O autor (2021)

Figura 24 - Comparação dos resultados *R-Squared*

Results comparison

```
data = [['Gradient boost', s1], ['Random Forest', s2]]
df = pd.DataFrame(data, columns = ['ML model', 'Accuracy'])
df
```

	ML model	Accuracy
0	Gradient boost	0.899303
1	Random Forest	0.729736

Fonte: O autor (2021)

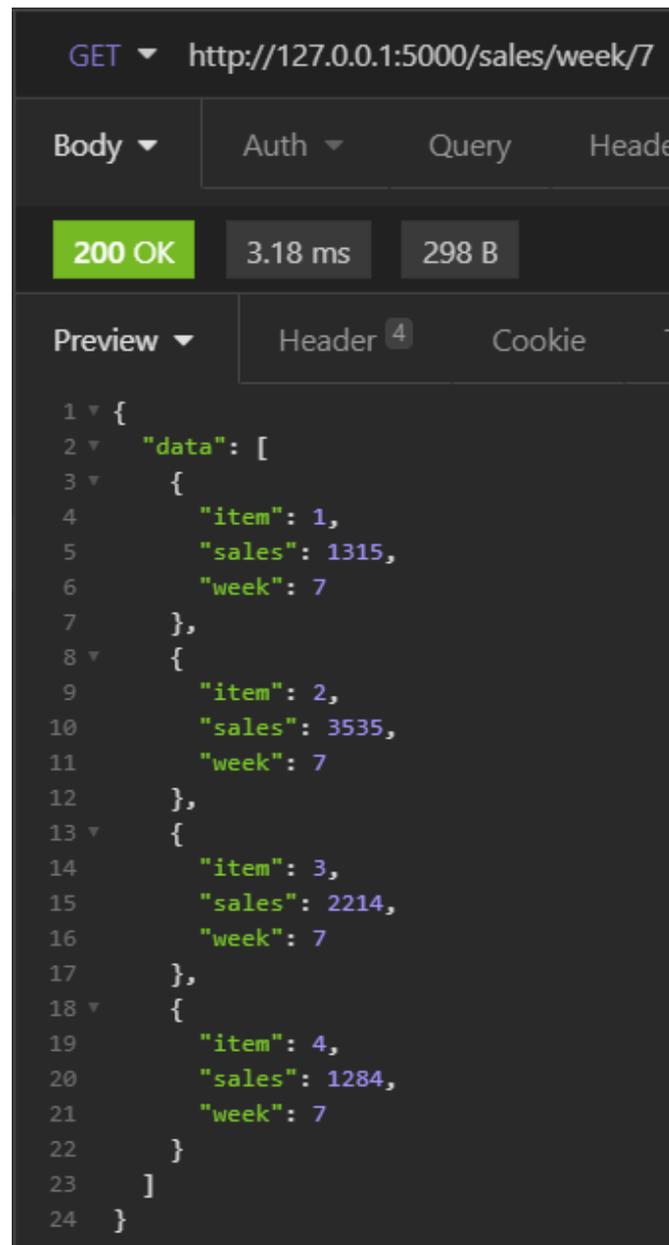
O modelo escolhido para ser utilizado na integração e cujos dados previstos foram enviados ao banco de dados utilizado foram os obtidos através da previsão realizada pelo algoritmo de *Gradient Boost Regressor* devido ao maior *R-squared Score*.

4.2 API REST

Peça chave para a integração entre cada parte do trabalho desenvolvido, a API cumpriu seu papel de disponibilizar os dados necessários para integração do modelo de inteligência artificial com a linha de manufatura.

A figura 25 mostra o resultado de uma requisição à API. A requisição espera como retorno os dados de previsão de demanda para a semana 7. Para os testes da API foi utilizado o software Insomnia, um cliente para realizar requisições utilizando métodos HTTP. A figura 26 mostra o mesmo tipo de requisição e a resposta no API Gateway desenvolvido no Node Red.

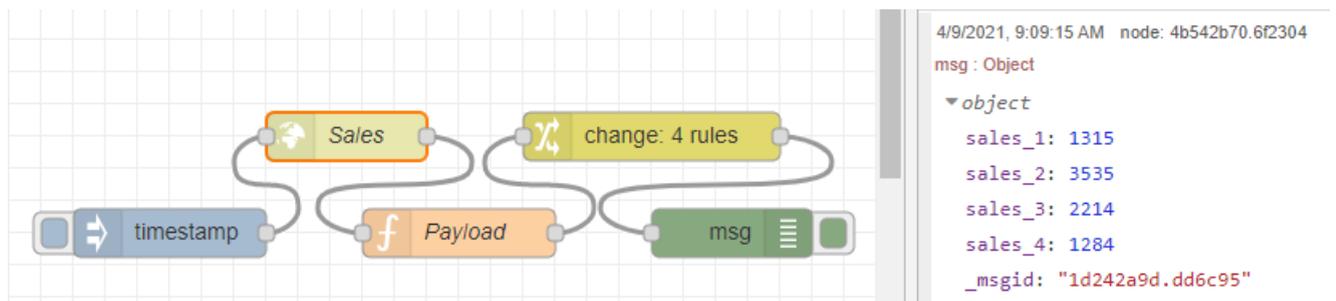
Figura 25 - Requisição à API no Insomnia



```
1 {
2   "data": [
3     {
4       "item": 1,
5       "sales": 1315,
6       "week": 7
7     },
8     {
9       "item": 2,
10      "sales": 3535,
11      "week": 7
12    },
13    {
14      "item": 3,
15      "sales": 2214,
16      "week": 7
17    },
18    {
19      "item": 4,
20      "sales": 1284,
21      "week": 7
22    }
23  ]
24 }
```

Fonte: O autor (2021)

Figura 26 - Requisição à API no Node Red

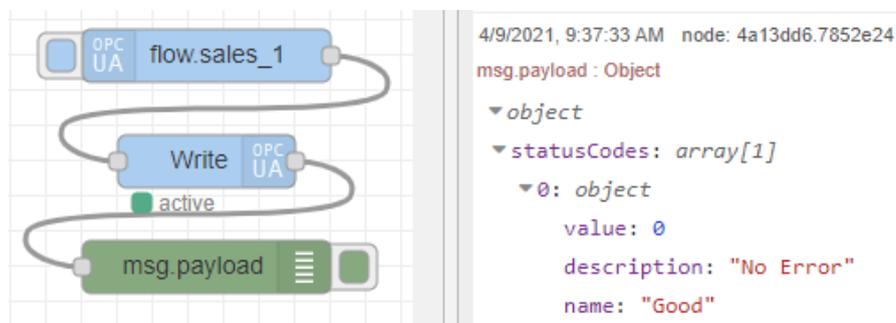


Fonte: O autor (2021)

4.3 Comunicação API Gateway e CLP

O resultado da comunicação entre o API Gateway e o CLP pode ser observado através dos logs de debug do Node Red ao lado dos valores recebidos pelo CLP. Como é possível observar na figura 27, o status da conexão retornou valor 0, com a descrição “No Error” indicando que os dados foram escritos com sucesso.

Figura 27 - Exemplo de resposta da escrita do dado de demanda prevista do item 1 na semana 7 via OPC UA



Fonte: O autor (2021)

E confirmamos esse sucesso através da observação dos valores das variáveis criadas para cada item no CLP, conforme a figura 28.

Figura 28 - Valores de demanda prevista para cada um dos itens na semana 7 no CoDeSys

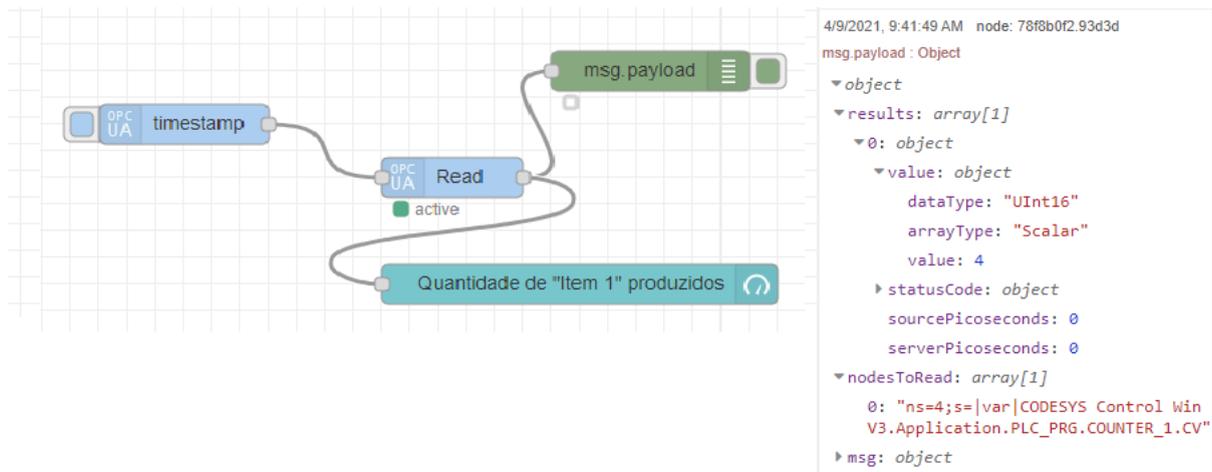
Device.Application.PLC_PRG					
Expression	Type	Value	Pre...	Ad...	Comment
SALES_1	INT	1315			
SALES_2	INT	3535			
SALES_3	INT	2214			
SALES_4	INT	1284			

Fonte: O autor (2021)

4.4 Comunicação entre CLP e Dashboard

Além da comunicação da API com o CLP é necessário enviar os dados da quantidade produzida de cada item em tempo de execução para o dashboard com o intuito de manter os gestores da linha de manufatura informados sobre o status da produção. Esse objetivo foi alcançado como pode ser visto na figura 29 por meio da observação dos logs de debug do Node Red em comparação com os valores presentes no CoDeSys apresentados na figura 30.

Figura 29 - Leitura da quantidade produzida do item 1



Fonte: O autor (2021)

Figura 30 - Leitura do contador no CLP

NAME	DATA TYPE	VALUE	DESCRIPTION
COUNTER_1	CTU		
CU	BOOL	FALSE	Count Up
RESET	BOOL	FALSE	Reset Counter to 0
PV	WORD	4	Counter Limit
Q	BOOL	FALSE	Counter reached the Limit
CV	WORD	3	Current Counter Value
SALES_1	INT	4	

Fonte: O autor (2021)

Os valores lidos do CLP são apresentados no dashboard desenvolvido que pode ser visualizado na figura 31.

Figura 31 - Exibição da quantidade produzida de cada item no dashboard



Fonte: O autor (2021)

Os valores para a quantidade de itens a serem produzidos exibidos no dashboard da figura 31 e nas figuras 29 e 30 são de uma simulação realizada em que as quantidades que deveriam ser produzidas de cada item foram reduzidas para executar o teste do sistema completo reduzindo o tempo necessário para finalização da produção dos itens. Portanto, se deve a isso o fato de os itens 1 e 2 estarem com sua produção completa com apenas 2 e 4 unidades, respectivamente.

5 CONCLUSÃO

Durante o desenvolvimento deste trabalho, foram apresentados e implementados os passos necessários para o desenvolvimento de um sistema completo de integração de inteligência artificial à uma linha de manufatura simulada, desde o desenvolvimento do modelo de previsão até a especificação de tecnologias e softwares que possuem ampla aplicabilidade em sistemas de produção no mercado industrial atual.

Foi possível utilizar diversos conhecimentos adquiridos ao longo do curso de Engenharia Mecatrônica em diferentes disciplinas, dentre eles destacam-se o uso de diferentes protocolos de comunicação industrial, a programação de APIs, a utilização de softwares de simulação, entre outros. Isso demonstra que a integração

dessas tecnologias estará cada vez mais presente no dia a dia dos engenheiros e valida a necessidade do profissional estar sempre atento aos avanços tecnológicos que vêm para tornar mais produtivo e facilitado a longo prazo o trabalho que é realizado por esse profissional.

Referente aos resultados do trabalho é possível afirmar que foram satisfatórios, uma vez que toda a integração proposta foi realizada. O sistema mostrou-se viável e funcional, trazendo à linha de manufatura um modelo que torna automatizada e mais assertiva a tomada de decisão, não esquecendo de deixar o usuário informado sobre o status da produção.

Em trabalhos futuros há a possibilidade de serem feitas melhorias pontuais em cada assunto apresentado ou então trabalhar de forma mais aprofundada em alguma das tecnologias desenvolvidas aqui.

Ao olhar do autor, um estudo de maior profundidade no desenvolvimento de um modelo de inteligência artificial mais elaborado, utilizando dados reais de mercado, levando em consideração outros indicadores para determinar a tomada de decisão e aprimorando os parâmetros utilizados na aplicação do algoritmo trariam uma grande evolução ao trabalho aqui desenvolvido. Dessa forma poderia ser validado em uma escala pequena em uma empresa.

REFERÊNCIAS

BAYAZIT, O., **Use of AHP in decision-making for flexible manufacturing systems**. Journal of Manufacturing Technology Management 16, Washington, 2005.

CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI. **Investimentos em indústria 4.0**. Disponível em: https://bucket-gw-cni-static-cms-si.s3.amazonaws.com/media/filer_public/8b/0f/8b0f5599-9794-4b66-ac83-e84a4d118af9/investimentos_em_industria_40_junho2018.pdf. Acesso em: 05 nov. 2020.

Dubois, P. F., **Python: Batteries Included, volume 9 of Computing in Science & Engineering**, 2007.

EXPERT.AI, **What is Machine Learning? A Definition**. Disponível em: <https://www.expert.ai/blog/machine-learning-definition/>. Acesso em: 12 fev. 2021.

FROST, Jim, **R-Squared**. Disponível em: <https://statisticsbyjim.com/glossary/r-squared/#:~:text=By%20Jim%20Frost,to%20the%20fitted%20regression%20line>. Acesso em: 20 mar 2021.

FUJIMAKI, **Five Challenges of Implementing AI in Manufacturing**. Disponível em: <https://industrytoday.com/five-challenges-of-implementing-ai-in-manufacturing/>. Acesso em: 15 fev. 2021.

GASSER, R. & FISCHER, K. & WAEFLER, T., **Decision Making in Planning and Scheduling: A Field Study of Planning Behaviour in Manufacturing**, Heidelberg, 2011.

GIROTTI, L. J.; MESQUITA, M. A. de. **Simulação e estudos de caso no ensino de planejamento e controle da produção: um survey com professores da engenharia de produção**. USP, Brasil. 176- 189p, 2016.

GUERRINI, F. M.; BELHOT, R.V.; AZZOLINI JÚNIOR, W. **Planejamento e controle da produção, modelagem e implementação**. Ed. Elsevier. 2. ed. Rio de Janeiro. 2019.

HARREL, C. R.; GHOSH, B. K.; BOWDEN, R., **Simulation Using ProModel®**. McGraw-Hill, 2000.

HEUER, Steffan. **Having an AI strategy is essential to survival**. Think:Act Magazine, Central Europe, ago. 2020. Disponível em: <https://www.rolandberger.com/sv/Point-of-View/Having-an-AI-strategy-is-essential-to-survival.html>. Acesso em: 01 dez. 2020.

KRAJEWSKI, L. J. & RITZMAN, L. P. (1999). **Operations management, strategy and analysis**, 5a Ed., Addison-Wesley, Reading, MA.

LUCKE D., CONSTANTINESCU C., WESTKÄMPER E. **Smart Factory - A Step towards the Next Generation of Manufacturing**. Springer, London, 2008. https://doi.org/10.1007/978-1-84800-267-8_23

MAKRIDAKIS, S., WHEELWRIGHT, S. C. & HYNDMAN, R. J. (1998). **Forecasting – methods and applications**, 3ª ed., John Wiley, New York.

MCCARTHY, J., MINSKY, M. L., ROCHESTER, N., & SHANNON, C. E., **A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence**, AI Magazine, 1955. <https://doi.org/10.1609/aimag.v27i4.1904>

MILMANN, K.J. & Avaizis, M., **Scientific Python volume 11 of Computing in Science & Engineering**, 2011.

MOHRI, Mehryar & ROSTAMIZADEH, Afshin & TALWALKAR, Ameet. **Foundations of Machine Learning**, Second Edition, Cambridge, 2018.

OPC FOUNDATION. **Classic.** Disponível em: <https://opcfoundation.org/about/opc-technologies/opc-classic/>. Acesso em: 07 mar. 2021.

OPC FOUNDATION. **Unified Architecture.** Disponível em: <https://opcfoundation.org/about/opc-technologies/opc-ua/>. Acesso em: 07 mar. 2021.

PEDREGOSA, F. & VAROQUAUX, G & GRAMFORT, A. & MICHEL, V. & THIRION, B. & GRISEL, O. & BLONDEL, M. & PRETTENHOFER, P. & WEISS, R. & DUBOURG, V. & VANDERPLAS, J. & PASSOS, A. & COURNAPEAU, D. & BRUCHER, M. PERROT, M. & DUCHESNAY, E., **Scikit-learn: Machine learning in Python.** Journal of Machine Learning Research, 2011.

POOLE, David & MACKWORTH, Alan. **Artificial Intelligence: Foundations of Computational Agents**, Cambridge, 2010.

PRIORE, P., PONTE, B., PUENTE, J., GÓMEZ, A., **Learning-based scheduling of flexible manufacturing systems using ensemble methods**, **Computers & Industrial Engineering**, 2018. doi: <https://doi.org/10.1016/j.cie.2018.09.034>

RADZIWON, A. & BILBERG, A. & BOGERS, M. & SKOV, E., **The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions**, *Procedia Engineering*, Volume 69, 2014, <https://doi.org/10.1016/j.proeng.2014.03.108>.

RASCHKA, Sebastian & MIRJALILI, Vahid, **Python Machine Learning: Machine Learning and Deep Learning With Python, Scikit-Learn, and TensorFlow 2**, Third Edition, Birmingham, 2019.

REDHAT. **O que é API REST?** Disponível em: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>. Acesso em: 03 mar. 2021.

REDHAT. **O que é API?** Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces> Acesso em: 03 mar. 2021.

SANTOS, R. F. D. **Estratégias Híbridas de Machine Learning e Simulação para a Resolução de Problemas de Escalonamento**. Dissertação de mestrado (Engenharia Eletrotécnica e de Computadores) - Universidade do Porto, Portugal, 2018.

SHALEV-SCHWARTZ, Shai & BEN-DAVID, Shai. **Understanding Machine Learning: From Theory to Algorithms** , Cambridge, 2014.

SITWARE. **O que é Indústria 4.0?** Disponível em: <https://www.siteware.com.br/metodologias/o-que-e-industria-4-0/>. Acesso em: 01 dez. 2020.

TRACTIAN. **O Impacto da Inteligência Artificial na Indústria Brasileira**. Disponível em: <https://traction.com/blog/o-impacto-da-inteligencia-artificial-na-industria-brasileira/>. Acesso em: 05 dez. 2020.

TVERSKY, A., & KAHNEMAN, D., **Judgment under uncertainty: Heuristics and biases**. Science, 1974.