



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CENTRO DE ENGENHARIAS
DEPARTAMENTO DE ENGENHARIA E CIÊNCIAS AMBIENTAIS
CURSO DE ENGENHARIA DE PRODUÇÃO

JHONATAN KAYO SILVA SENA

**DESENVOLVIMENTO DE MODELO DE REVISÃO PERIÓDICA DE ESTOQUES
UTILIZANDO SIMULAÇÃO COMPUTACIONAL E INTERAÇÃO POR MEIO DE
INTERFACE GRÁFICA**

MOSSORÓ - RN
2023

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

SS474 SILVA SENA, Jhonatan Kayo.
d DESENVOLVIMENTO DE MODELO DE REVISÃO PERIÓDICA
DE ESTOQUES UTILIZANDO SIMULAÇÃO COMPUTACIONAL E
INTERAÇÃO POR MEIO DE INTERFACE GRÁFICA /
Jhonatan Kayo SILVA SENA. - 2023.
43 f. : il.

Orientador: David Custódio de Sena.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Engenharia de
Produção, 2023.

1. Gestão de estoques. 2. Simulação discreta.
3. Revisão periódica. 4. SimPy. I. de Sena, David
Custódio, orient. II. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade
com AACR2 e os dados fornecidos pelo(a) autor(a).
Biblioteca Campus Mossoró / Setor de Informação e Referência
Bibliotecária: Keina Cristina Santos Sousa e Silva
CRB: 15/120

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JHONATAN KAYO SILVA SENA

**DESENVOLVIMENTO DE MODELO DE REVISÃO PERIÓDICA DE
ESTOQUES UTILIZANDO SIMULAÇÃO COMPUTACIONAL E INTERAÇÃO
POR MEIO DE INTERFACE GRÁFICA**

Trabalho de Conclusão de Curso
apresentado ao curso de Engenharia de
Produção da Universidade Federal Rural
do Semiárido – UFERSA, como
requisito para obtenção do título de
Bacharel em Engenharia de produção.

Defendida em: 18 / 05 / 2023.

BANCA EXAMINADORA

Prof. Dr. David Custódio de Sena
Presidente

Prof^a. Dr^a. Miriam Karla Rocha
Membro Examinador

Prof. Dr. Cristiano Luis Turbino de Franca e Silva
Membro Examinador

MOSSORÓ – RN
2023

DEDICATÓRIA

Dedico este trabalho a todos que, de alguma forma me apoiaram nessa jornada, em especial a minha família, que me incentivou desde o princípio.

AGRADECIMENTOS

Agradeço a minha família, que me deu toda estrutura e apoio nos momentos bons e ruins de minha vida, para que continuasse trilhando o meu sonho em se tornar engenheiro. Nas maiores adversidades e dificuldades estiveram comigo para me levantar e seguir em frente, mostrando o significado do que família, efetivamente, é. Por família, especificamente, agradeço meu tio Jadson, minha avó Marli e meu avô Jadiomar, que me criaram e me deram todas as condições e ensinamentos que me tornaram quem hoje sou.

A meu companheiro Gustavo por estar comigo nos bons e maus momentos, por me dar forças para continuar quando eu não mais acreditava que conseguiria e me incentivar a concluir essa etapa de minha vida.

A meus amigos e colegas, em especial Jéssica e Adlan, que me fazem acreditar no lado bom da amizade e fraternidade. Agradeço também por me mostrarem formas diferentes de ver a vida e o mundo.

Ao meu professor e orientador David pela compreensão e paciência em relação a todas as dificuldades que tive no andamento do trabalho. Por me incentivar a explorar as possibilidades e não me satisfazer com o trivial.

Ao meu professor Thomas, que me incentivou além da faculdade e me fez crescer como profissional.

Por fim agradeço a Deus ou a força que rege esse mundo, pelas condições de executar esse trabalho, de ter seguido nesse curso, de ter tido condição, conhecimento e até um pouco de sabedoria na jornada que é o tornar-se engenheiro de produção.

“Não devemos nos questionar porque algumas coisas nos acontecem e sim o que podemos fazer com o tempo que nos é dado”.

(J. R. R. Tolkien)

RESUMO

Modelos de simulação discreta são amplamente utilizados na resolução de problemas complexos ou que envolvam uma análise em um período superior ao que é viável o homem analisar manualmente ou por ferramentas menos robustas. A gestão de estoques tem ganhado cada vez mais destaque no mundo, em especial no que tange a cadeia de suprimentos, dada a complexidade a nível global que as relações comerciais ocorrem. Esse trabalho tem como objetivo introduzir a simulação discreta a gestão de estoques, utilizando do modelo de revisão periódica em um modelo computacional simplificado que possibilite ao usuário interagir com um modelo de simulação discreta sem necessariamente compreendê-la. A solução desenvolvida busca oferecer, de forma gráfica, a visualização do nível de estoque e como os parâmetros de estoque interagem entre si no modelo de revisão periódica, otimizando assim a compreensão de um dos pilares da gestão de estoque, o momento de pedir. Os resultados do modelo de simulação e interface gráfica desenvolvidos foram validados por meio de aplicação em cenário obtido de estudo acadêmico e demonstram que a ferramenta desenvolvida auxilia de maneira satisfatória no entendimento sobre revisão periódica de estoques, contribuindo no campo da gestão de estoques em relação a interfaces onde é possível interagir com os parâmetros do estoque e obter como resposta feedback visual.

Palavras-chave: Gestão de estoques, simulação discreta, revisão periódica, SimPy.

ABSTRACT

Discrete simulation models are widely used to solve complex problems or those that require analysis over a period of time that is not feasible for humans to analyze manually or with less robust tools. Inventory management has gained increasing attention worldwide, particularly in the supply chain, given the complexity of global trade relationships. This work aims to introduce discrete simulation to inventory management, using the periodic review model in a simplified computational model that enables users to interact with a discrete simulation model without necessarily understanding it. The developed solution seeks to provide a graphical visualization of the inventory level and how the inventory parameters interact with each other in the periodic review model, thus optimizing the understanding of one of the pillars of inventory management, the reorder point. The results of the simulation model and graphical interface developed has been validated through application in a real-world scenario obtained from an academic study and demonstrate that the developed tool satisfy the understanding about periodic stock review, contributing on the area of inventory management in terms of a interface where it is possible to interact with inventory parameters and obtain visual feedback as response.

Keywords: Inventory management, discrete simulation, periodic review, SimPy.

LISTA DE FIGURAS

Figura 01. Finalidade básica do estoque	15
Figura 02. Interface do SCOPS	16
Figura 03. Etapas do projeto	21
Figura 04. Comunicação entre cliente e servidor via fetch API	22
Figura 05. Iteração da simulação e servidor via Threading	24
Figura 06. Campos de alimentação da simulação	25
Figura 07. Gráfico da simulação	26
Figura 08. Alteração de parâmetros da simulação em vermelho	26
Figura 09. Solicitação de consumo do depósito	27
Figura 10. Momento de solicitar ressuprimento do estoque	28
Figura 11. Acesso ao nível de estoque via JSON em servidor HTTP	28
Figura 12. Validação do modelo com exemplo aplicado	29

LISTA DE TABELAS

Tabela 01. Elementos de um modelo computacional	14
Tabela 02. Heurísticas de Nielsen	19
Tabela 03. Elementos da simulação	23

SUMÁRIO

1. INTRODUÇÃO	12
1.1 OBJETIVOS.....	13
2. REVISÃO DA LITERATURA	14
2.1 SIMULAÇÃO A EVENTOS DISCRETOS	14
2.2 SIMULAÇÃO EM GESTÃO DE ESTOQUES	15
2.2.1 Determinação do tamanho do lote.....	16
2.1.2 Demanda constante	17
2.1.3 Lote econômico de compra.....	17
2.1.4 Lote econômico de produção.....	17
2.2 DETERMINAÇÃO DO PERÍODO DE COMPRA	18
2.2.1 Reposição contínua.....	18
2.2.2 Reposição periódica.....	18
2.3 HEURÍSTICA E USABILIDADE.....	19
3 PROCEDIMENTOS METODOLÓGICOS	21
3.1 CLASSIFICAÇÃO DA PESQUISA.....	21
3.2 ETAPAS DA PESQUISA.....	21
3.3 FERRAMENTAS UTILIZADAS	22
3.3.1 SimPy.....	22
3.3.2 Fetch API.....	22
3.4 MODELAGEM COMPUTACIONAL DA SOLUÇÃO	23
3.5 INTERFACE HOMEM-MÁQUINA.....	24
4 RESULTADOS E DISCUSSÕES	27
5 CONSIDERAÇÕES FINAIS	30
5.1 PERSPECTIVAS DE TRABALHOS FUTUROS.....	31
REFERÊNCIAS	32
APENDICE A – MODELO COMPUTACIONAL DE SIMULAÇÃO	35
APENDICE B – INTERFACE WEB.....	39

1. INTRODUÇÃO

A gestão de estoques é um dos maiores desafios enfrentados pelas empresas em todo o mundo. Uma gestão inadequada pode levar a problemas como excesso de estoque, falta de produtos em estoque, aumento de custos e diminuição da qualidade do atendimento ao cliente (LAWRENCE; CHIROMO, 2020). Nesse sentido, a simulação discreta tem se mostrado uma técnica promissora para a gestão de estoques, permitindo a criação de modelos que reproduzem o comportamento dos estoques ao longo do tempo e auxiliam na tomada de decisões (DUONG, 2017).

Estudo da KPMG indicou que em 2021 o varejo brasileiro acumulou cerca de 24 bilhões de reais em perdas, parte destas relacionadas ao controle de estoque, mais que isso, parte das perdas foi decorrente da não observância de fatores como validade e armazenamento dos produtos. O uso de tecnologia foi uma das formas sugeridas para FecomercioSP para contenção dos danos causados pela ausência de uma política adequada de gestão de estoques (FECOMERCIO, 2022).

Na indústria, dentro do âmbito do gerenciamento da cadeia de suprimentos, o uso de simulação já é aplicado integrado a inteligências artificiais (FUJIMAKI, 2023), que demonstra não só haver a necessidade como grande relevância no desenvolvimento de tecnologias aplicadas a área dentro do campo acadêmico e empresarial. O estudo indica ainda que 42% das empresas tecnologicamente avançadas na área apresentam crescimento na ordem de dois dígitos. Artigo de Inman (1999), já em 1999, aponta outras variáveis que reforçam a importância de uma maior assertividade na gestão de estoques nas organizações: a gestão ambiental, fator cada vez mais presente nas empresas pelo mundo, o autor cita que o desperdício e ingerência no planejamento e controle da produção podem afetar profundamente a cadeia de suprimentos neste novo cenário que se desenha.

Além de todos os fatores econômicos relacionados ao desenvolvimento de ferramentas tecnológicas voltadas a gestão, periódicos como o de Vavra *et al.* (2011) apontam a importância do desenvolvimento dessas tecnologias também para o ensino, o estudo aponta haver consenso acadêmico a respeito da potencialização do aprendizado em situações onde as ferramentas estáticas de ensino são substituídas por ferramentas dinâmicas e que o feedback imediato e visualizações geradas por computador aumentam a atenção e compreensão dos alunos sobre assuntos diversos, desde que estas ferramentas sejam utilizadas junto de material teórico adequado, sendo complementar ao aprendizado tradicional.

Vera *et al.* (2007) corrobora os apontamentos do artigo anteriormente citado demonstrando em estudo o leque crescente de ferramentas computacionais orientadas ao aprendizado e como ferramentas que possibilitam interação em tempo real maximizam o entendimento, em especial de pessoas com dificuldades de aprendizado.

1.1 Objetivos

Dado a problemática apresentada e considerando a tendência no desenvolvimento de ferramentas computacionais voltadas a gestão ou ensino o trabalho tem como objetivo geral desenvolver ferramenta interativa e gráfica que auxilie no entendimento sobre o gerenciamento de estoques, utilizando de simulação computacional que possibilite a interação, em tempo real, com um fluxo de consumo e ressuprimento de estoque, em ambiente simulado.

Para os objetivos específicos do trabalho, vislumbrando as possíveis etapas necessárias ao atingimento do objetivo geral do estudo, tem-se os seguintes tópicos:

- Criar modelo de simulação a eventos discretos sobre gerenciamento de estoques com reposição periódica;
- Criar uma interface gráfica que possibilite interação homem-máquina de forma simplificada e intuitiva;
- Desenvolver um ambiente que permita a interação entre o modelo computacional de simulação e a interface homem-máquina.

2. REVISÃO DA LITERATURA

2.1 Simulação a eventos discretos

A simulação tem tido destaque nos últimos tempos pela finalidade de sua aplicação, que é simular um cenário real em um ambiente projetado (HILLIER; LIEBERMAN, 2013). A possibilidade de simular cenários reais possibilita a quem estuda estes cenários obter uma resposta semelhante ao real em ambiente projetado com custo significativamente menor, fazendo uso de estatística e modelos de sistema estocásticos.

O uso da simulação envolve alguns elementos dos quais é importante ter conhecimento dos conceitos para a modelagem de um sistema (SOUZA, 2009), podemos conferir esses elementos na Tabela 01:

Elemento	Definição
Entidades	São as partes que se movimentam dentro do modelo de simulação, elas são dotadas de características que as permitem interagir de forma única com o sistema.
Atributos	Características que definem uma entidade, em termos práticos, se definirmos uma entidade chamada “carro”, o atributo “velocidade” definiria uma de suas características, que é a velocidade que o mesmo se desloca a cada passo no sistema;
Recursos	podem ser considerados entidades, porém estes não se deslocam pelo sistema, mas sim alimentam as entidades de acordo com a modelagem definida para si;
Variáveis de estado	Representam características de um sistema. O número de usuários em uma fila por exemplo é uma variável de sistema, pois representa o todo, enquanto o usuário, individualmente, representa uma entidade;
Eventos	O evento pode ser caracterizado como a etapa em que o sistema sofre alguma alteração que provoca mudança no estado do sistema como um todo. Os eventos são diretamente relacionados ao resultado das variáveis de estado.

Tabela 01. Elementos de um modelo computacional, adaptado de SOUZA (2009).

2.2 Simulação em gestão de estoques

Correa *et al.* (2001) define “estoque” como um elemento gerencial necessários as organizações para manter sua capacidade produtiva, sendo apenas recentemente mais plenamente compreendido quanto a sua aplicabilidade organizacional. O estoque representa uma reserva de materiais necessários a operação da empresa, seja esta direcionada a comercialização de produtos ou serviços, e, portanto, tem como objeto de estudo o ponto ideal de manutenção dentro de uma empresa. Paoleschi (2019) complementa, entretanto, uma mudança de paradigma quanto ao conceito mais tradicional de gestão de estoque, citando a importância com o advento da indústria 4.0 do uso de tecnologias como a internet das coisas e inteligência artificial visando um controle mais efetivo e eficiente dos estoques na cadeia de suprimentos global.

O estoque pode ser compreendido entre duas etapas de um processo, o suprimento, pois a matéria ou produto, dentro do processo produtivo, é fornecida dentro de uma cadeia logística interna ou externa, e o consumo, no qual o material ou produto que foi suprido é utilizado com a finalidade de oferecer o produto ou serviço a população consumidora deste, conforme figura 01:

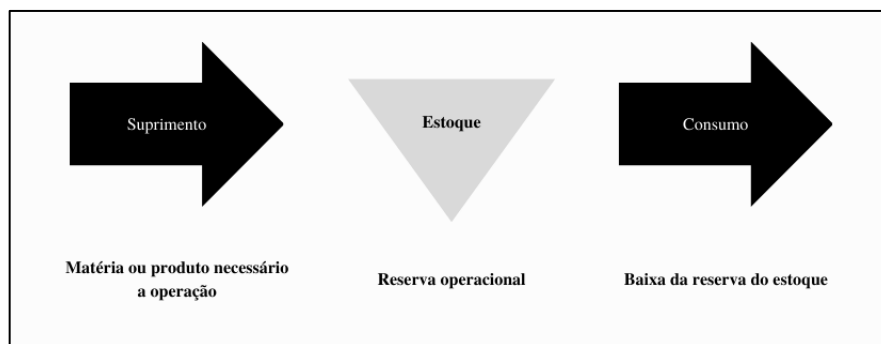


Figura 1. Finalidade básica do estoque – Adaptado de CORREA (2001).

Longo e Mirabelli (2009) apontam que dentro do atual estado da arte da simulação discreta, há diversos softwares generalistas que possibilitam a estudiosos simular situações para objetivos gerais, mas que não são intuitivos ou específicos o suficiente para usabilidade direcionada, como a gestão de estoques. No estudo é apresentado desde as plataformas de simulação discreta mais utilizadas do mercado, como o AnyLogic e Arena, como também uma plataforma desenvolvida em C++ e específica para a gestão de estoques na cadeia de suprimentos, o SCOPS, sigla para Supply Chain Order Performance Simulator, a plataforma

permite controlar, via interface gráfica, desde o tempo da simulação até o gerenciamento de estoque do CD, lojas e fornecedores, como pode ser visto na Figura 2.



Figura 02. Interface do SCOPS (LONGO; MIRABELLI, 2009)

Holden (2017) segue uma abordagem semelhante desenvolvendo um modelo de simulação utilizando a biblioteca SimPy, a validação do modelo segundo o autor, baseou-se em comparar o cenário simulado com a teoria da gestão de estoques, visualizando se a distribuição normal da demanda gera um cenário adequado ao que a teoria sugere.

Para avançarmos dentro do que é necessário a gestão de estoques na simulação, portanto, é necessário compreender os principais tópicos teóricos relacionados ao objeto de estudo, Junior (2019) separa a gestão de estoque em “Determinação de tamanho do lote” e “Níveis de estoque e momento de pedir”.

2.2.1 Determinação do tamanho do lote

Junior (2019) classifica os métodos para determinação de tamanho de lote em dois tipos: os adequados a demanda constante e os adequados a demanda variável ao longo do período desejado. Os métodos para mensuração do tamanho adequado de lote serão abordados nos tópicos subsequentes.

2.1.2 Demanda constante

Os métodos mais recomendados a demanda constante pelos itens do estoque dentro de um período estipulado são o LEC, lote econômico de compra, e LEP, lote econômico de produção (JUNIOR, 2019).

2.1.3 Lote econômico de compra

Tomemos os parâmetros/variáveis necessários ao cálculo do LEC como os seguintes:

- c: Custo unitário (R\$/unidade);
- i: Custo de estocagem anual (% por ano);
- h (i x c): Custo de estocagem anual (R\$ por unidade por ano);
- A: Custo do pedido, independentemente da quantidade pedida (R\$/pedido);
- D: Demanda por unidade de tempo.

A partir destes parâmetros, é possível calcular o lote econômico de compra através da Equação 1:

$$Q * = \sqrt{\frac{2 \cdot A \cdot D}{h}} \quad (1)$$

Equação 1. Cálculo do lote econômico de compra, adaptado de JUNIOR (2019)

2.1.4 Lote econômico de produção

Tomemos os parâmetros/variáveis necessários ao cálculo do LEP como os seguintes:

- c: Custo unitário (R\$/unidade);
- i: Custo de estocagem anual (% por ano);
- h (i/c): Custo de estocagem anual (R\$ por unidade por ano);
- A: Custo de setup, independentemente da quantidade produzida (R\$/setup);
- D: Demanda por unidade de tempo;
- u: taxa de produção.

A partir destes parâmetros, é possível calcular o lote econômico de produção através da Equação (2):

$$Q * = \sqrt{\frac{2 \cdot A \cdot D}{h(1 - \frac{D}{u})}} \quad (2)$$

Equação 2. Cálculo do lote econômico de produção, adaptado de JUNIOR (2019).

2.2 Determinação do período de compra

Além do tamanho do pedido, é importante também avaliar a periodicidade de compra para o manutenção do estoque, com o mínimo de prejuízo planejado. Dentre os vários métodos relacionados ao ressuprimento de estoque, Ching (2010) apresenta os métodos de reposição contínua e periódica de estoques.

2.2.1 Reposição contínua

O método de reposição contínua, em um cenário onde busca-se reduzir os custos de volume de inventário, tem como princípio avaliar *quando* solicitar um novo lote. Visando manter o menor volume de estoque possível sem ter a produção ou prestação de um serviço impactada por essa estratégia.

2.2.2 Reposição periódica

Na reposição periódica temos corrigida uma problemática que ocorre no método de reposição contínua, que é a previsibilidade de pedido. Inverte-se a análise de unidades em estoque para unidades de tempo na consideração do período de pedir a reposição do estoque, conforme Equação (3):

$$T = \frac{Q}{D \cdot P} \cdot P$$

Equação 3. Período de revisão. Adaptado de Ching (2010).

Sendo:

Q : Tamanho do lote,

D : Demanda por período,

P : Período.

Na reposição periódica o tamanho do pedido também difere da revisão contínua quanto a sua forma, este é calculado pela diferença do nível do estoque no momento do pedido quando a seu estoque máximo (BEZERRA, 2020). O nível de estoque máximo pode ser calculado de acordo com a equação:

$$Em = D \cdot (P + L) + Es$$

Equação 4. Cálculo do estoque máximo. Adaptado de (BEZERRA, 2020).

Onde:

- **Em:** Estoque máximo;
- **D:** demanda média;
- **P:** Período de revisão;
- **L:** Tempo de entrega;
- **Es:** Estoque de segurança.

2.3 Heurística e usabilidade

No que tange ao uso de um sistema, há diversos estudos na área da ergonomia que indicam da importância da usabilidade da interface utilizada, seja esta física ou virtual (GOBBO et al., 2017).

As heurísticas de Nielsen foram criadas visando solucionar problemas de usabilidade relacionados a interfaces, sugerindo, em 10 tópicos, fundamentos básicos de usabilidade em interfaces virtuais (CORDEIRO et al., 2009). As 10 heurísticas de Nielsen podem ser vistas na tabela 02:

Heurística	Descrição
1. Diálogo simples e natural	O status do sistema deve ser visível e comum ao ser humano.
2. Linguagem do usuário	O sistema deve conversar com o mundo real, um botão deve parecer um botão.
3. Minimiza a memorização do usuário	O sistema deve resolver situações de erro de forma simplificada.
4. Consistência	Padrões em sistemas, assim como no mundo real, potencializam a curva de aprendizado.
5. Feedback	A interação com o sistema deve ser visualmente respondida.
6. Encerramento claro	A interface do sistema deve deixar claro o que aquele elemento executa.
7. Atalho	A interface deve ser clara para usuários novos e livre para usuários experientes.
8. Mensagem de erro clara	Um erro deve ser sinalizado propriamente pelo que ele é, não por um código de máquina.
9. Prevenção de erros	Os elementos de uma interface que não podem ser usados devem ser escondidos ou desativados.

10. Ajuda e documentação	Soluções complexas devem ser documentadas.
--------------------------	--

Tabela 02. Heurísticas de Nielsen. Adaptado de (GOBBO et al., 2017; NIELSEN, 1994).

3 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo será apresentada a metodologia utilizada na solução da problemática abordada.

3.1 Classificação da pesquisa

Para Andrade (2012), o mínimo que uma pesquisa ter quanto a sua classificação é a definição de sua natureza, objetivo, procedimentos e seu objeto (de pesquisa). A resolução da problemática será feita em ambiente computacional, aplicando conceitos da simulação de eventos discretos e gestão de estoques.

Pode-se classificar a natureza da pesquisa deste trabalho como aplicada, pois todos os conceitos levantados durante o trabalho apresentado serão implementados em um ambiente que visa simular um cenário real de curva de estoque.

Quanto aos objetivos, podemos classifica-la como exploratória, dado o objetivo de validar as informações levantadas na resolução da problemática abordada, ao mesmo tempo que busca servir como base para maior elucidação desta aqueles que busquem evoluir a temática apresentada.

3.2 Etapas da pesquisa

Dado o objetivo do trabalho em desenvolver ferramenta que possibilite visualizar uma curva de estoque simulada e virtualmente manipulada, aplicando simulação discreta, o trabalho será dividido em quatro etapas: Revisão bibliográfica, modelagem da solução, aplicação computacional e análise dos resultados.

Na revisão bibliográfica será compreendido os conceitos que englobam tanto a simulação de eventos discretos quanto os métodos de gestão do estoque, além de levantar material científico relacionado a ferramenta que será utilizada na etapa da aplicação computacional, a biblioteca SimPy da linguagem de programação computacional Python.

Posterior a revisão bibliográfica, a modelagem do problema será desenvolvida em fluxograma para posterior aplicação computacional considerando os padrões de linguagem e da biblioteca de simulação referida anteriormente. Com o modelo já implementado computacionalmente será avaliado os resultados da pesquisa e aplicação, apresentando-se as considerações finais do referido projeto.

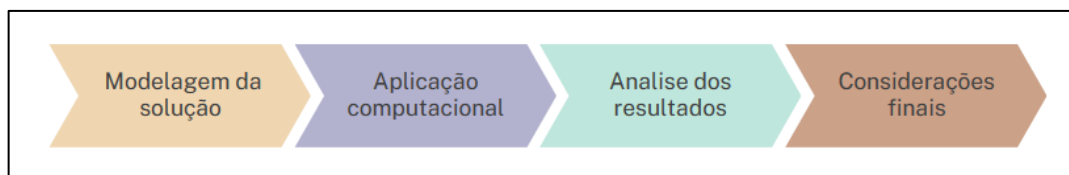


Figura 03. Etapas do projeto (Fonte: Autoria própria).

3.3 Ferramentas utilizadas

Para tal será utilizada a biblioteca SimPy, desenvolvida em Python, para o *backend* da interface, que executará a simulação. Para a captação dos dados para exibição no *frontend*, será utilizado o método *fetch*, uma API da linguagem Javascript que permite a interceptação de dados e interação com servidores através de requisições HTTP.

Além das ferramentas acima, que são base para os objetivos específicos do estudo, serão utilizadas também algumas bibliotecas complementares: *threading*, *numpy*, *simpy.rt*, *json* e *flask* no python e a biblioteca *chart.js* no Javascript, para exibição gráfica da curva de nível de estoque.

3.3.1 SimPy

A documentação do SimPy (2023) o classifica como um framework de simulação de eventos discretos orientado a processos. Os processos no SimPy são definidos pelas *generator functions* (PYTHON FOUNDATION, 2023), que itera o estado do sistema na simulação, permitindo guardar seu estado e os dados gerados a partir da retenção deste estado.

3.3.2 Fetch API

A Fetch API é uma ferramenta criada para uso junto a linguagem Javascript visando possibilitar o acesso e manipulação das informações fornecidas por um servidor utilizando de requisições e respostas através do método HTTP (MILLS *et al.*, 2023). A necessidade de uso da ferramenta se dá devido a comunicação entre sistemas com linguagens distintas ser possível somente utilizando de métodos alternativos.

Dentro do contexto deste estudo, a Fetch API possibilita a interação entre o que é gerado na simulação, do lado do servidor, e o que é exibido ao usuário, do lado do cliente.

A Figura 4 ilustra de forma simplificada a lógica dessa comunicação, onde a interação é feita do lado do cliente, na então interface homem-máquina, a resposta da solicitação vem do

lado do servidor, porém toda essa interação é controlada por uma API no meio do fluxo, a Fetch API.

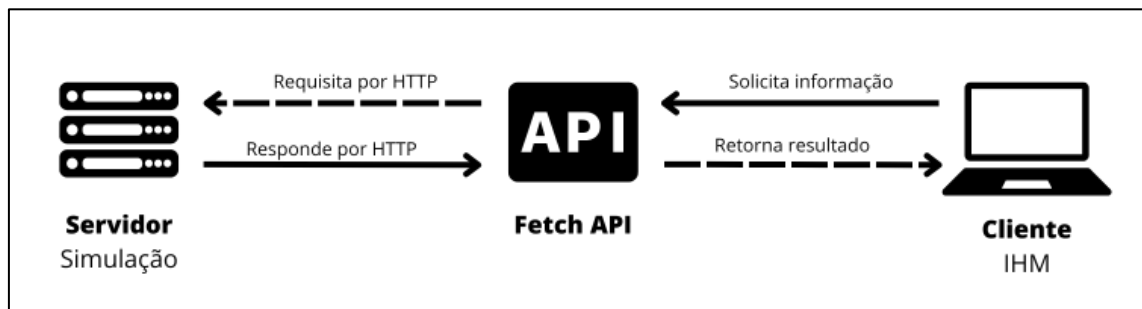


Figura 04. Comunicação entre cliente e servidor via Fetch API (Autoria própria).

3.4 Modelagem computacional da solução

Dentro do escopo da solução ao problema proposto, que seria uma interface onde é possível visualizar e interagir com o processo de gestão do estoque, é importante definir os parâmetros da solução. Utilizando como base o referencial teórico levantado, foi observado que a simulação utiliza de algumas entidades e recursos para funcionar.

Foi escolhido um método de gestão de estoques para determinação do período de compras, o modelo de revisão periódica.

Dos elementos do sistema, considerando o que é necessário ao funcionamento de um modelo de revisão periódica, foi definido os parâmetros demonstrados na tabela 03.

Nome	Tipo	Descrição	Atributos
Estoque	Entidade	Variável onde é armazenado o estado do estoque após cada interação.	- Estoque inicial; - Estoque máximo.
Gerir estoque	Recurso	Determina se há a necessidade de realizar pedido e quantidade demandada ao estoque em uma etapa específica da simulação.	- Período de revisão; - Demanda (segue distribuição normal).
Fazer pedido	Evento	Acionado ao atingir o período de revisão. É feito o cálculo do estoque máximo da entidade estoque menos o estoque atual.	

Depósito	Variável do sistema	Estado global do estoque.	• Nível do estoque; • Perdas
----------	---------------------	---------------------------	---

Tabela 03. Elementos da simulação (Fonte: Autoria própria).

Aplicado os parâmetros e lógica do recurso de gestão de estoque ao script do SimPy, foi implementado simultaneamente a execução da simulação via python, possibilidade da atualização dos parâmetros e obtenção do estado geral da simulação via servidor. Para isso foi necessário implementar outras duas bibliotecas ao modelo computacional de simulação do SimPy, a biblioteca *Threading*, necessária para execução multiprocessos em um script e a biblioteca *Flask*, que permite executar e recuperar informações de um script python utilizando um servidor http. Ao fim da implementação, o script possuía uma camada principal, gerenciada pelo *Threading*, que o mantinha sempre em espera de um novo comando e duas camadas auxiliares, a própria simulação via simpy e o servidor flask, que poderia inicia-la ou manipulá-la via comandos http. Essa interação pode ser melhor visualizada na figura 5.

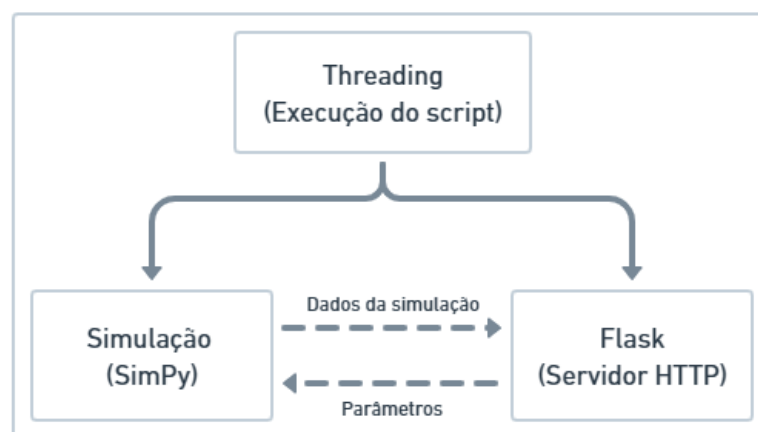


Figura 05. Interação da simulação e do servidor via threading. (Autoria própria)

3.5 Interface homem-máquina

A fim de diferenciar e agregar a compreensão sobre a gestão de estoques por meio de simulação computacional auxiliada pela biblioteca SimPy, é necessário o desenvolvimento de uma interface que possibilitasse a quem gerenciasse a simulação visualizar o nível de estoque e alterar seus parâmetros. Com a evolução da WEB e dos dispositivos móveis, é interessante

que a solução possa não só rodar em rede, mas em tempo real. Como a execução da simulação ocorria via servidor http, se faz necessário o desenvolvimento de uma interface web que, fazendo uso da fetch API do Javascript, encaminhasse requisições http ao servidor onde a simulação estivesse ocorrendo.

A interface, dentro do objetivo abordado, deve atender alguns requisitos.

- Ser visualmente tátil, possibilitando ao gestor da simulação assimilar onde os comandos serão enviados e onde as respostas serão retornadas.
- Retornar, em tempo real, e de forma visual, o nível de estoque.

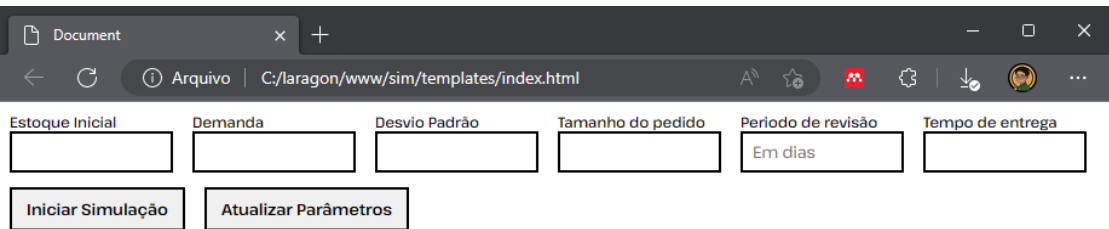


Figura 06. Campos de alimentação da simulação (Autoria própria).

Como as heurísticas de Nielsen (1994) sugerem, a interface, independente do escopo ser real ou virtual, deve prevenir erros de seu usuário, portanto, para iniciar a simulação, o usuário deve primeiramente alimentar algumas variáveis globais e comuns ao modelo de revisão periódica de estoque, são elas: Estoque inicial, demanda, desvio padrão (normal), tamanho do pedido, período de revisão e tempo de entrega.

Ao informar essas variáveis e clicar no botão “iniciar simulação”, a mesma será executada no servidor, iniciando a classe Depósito do simpy, que conforme tabela 03 é o próprio ambiente no qual a simulação de revisão periódica de estoque ocorre. O resultado dessa interação entre a interface homem máquina e o servidor pode ser visto na figura 07.

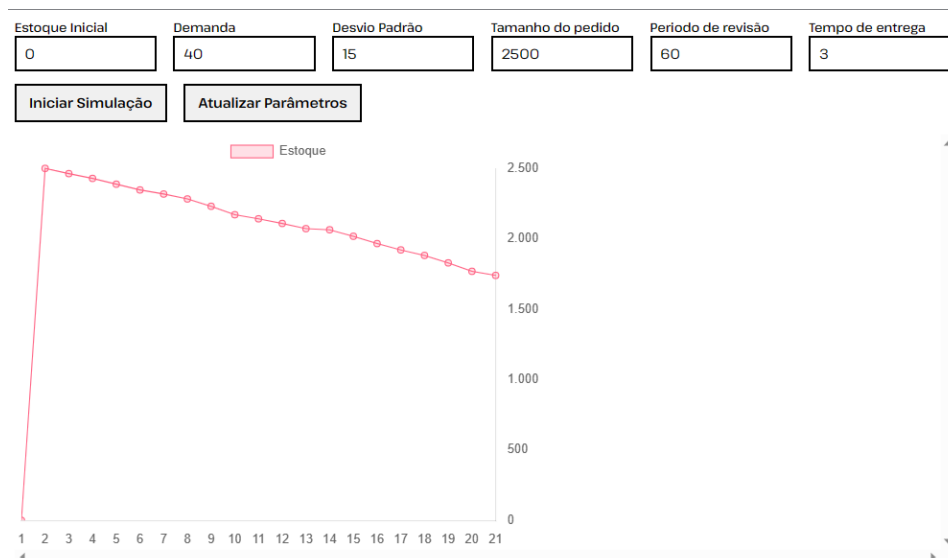


Figura 07. Gráfico da simulação (Autoria própria).

Quaisquer outras variáveis que não sejam o estoque inicial podem ser alteradas no decorrer da simulação, através do botão atualizar parâmetro. Se atualizarmos a demanda de estoque para “100”, por exemplo, poderemos notar, em tempo real, uma drástica mudança na curva de estoque durante o tempo de execução, conforme figura 08.

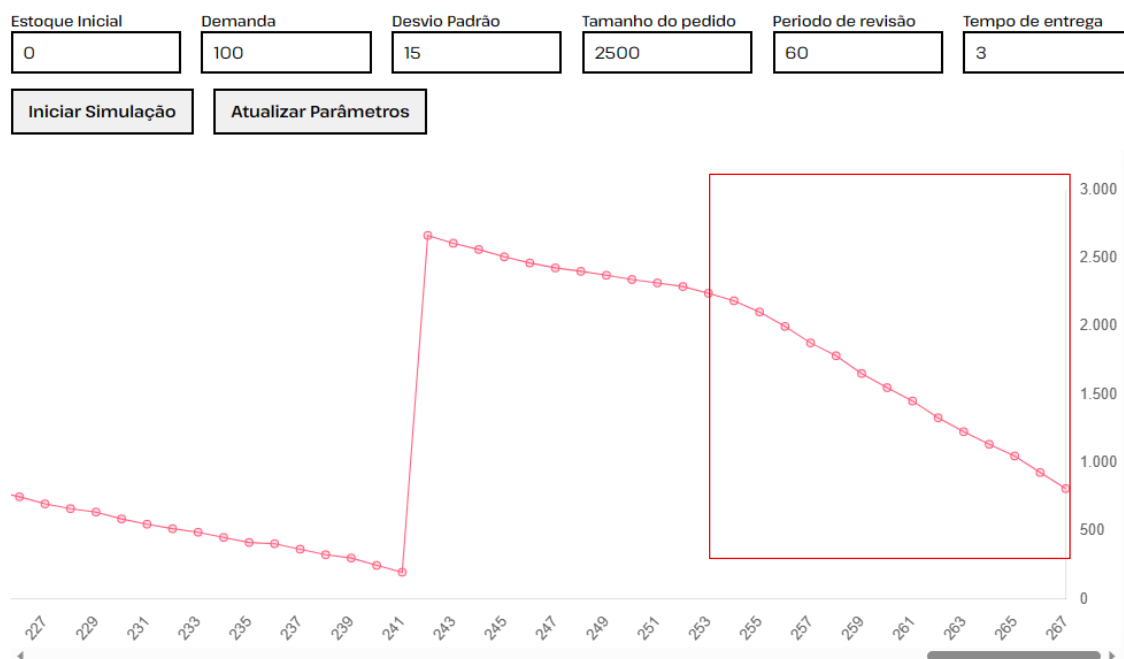


Figura 08. Alteração de parâmetros da simulação em vermelho (Autoria própria).

Uma dificuldade relacionada a interface homem-máquina em relação a visualização dos dados estava na duração da simulação, quanto maior a simulação, mais achatado seria o gráfico exibido. A solução foi implementada através de programação, onde seria calculado o tamanho

do gráfico em pixels, que é uma unidade de medida para telas, e ao invés de achatar o gráfico para exibi-lo integralmente, o gráfico seria exibido a partir de seu último pixel, mantendo-o sempre atualizado.

A aplicabilidade desse modelo computacional, junto de sua interface de comando, está na possibilidade de avaliar cenários através da alimentação desses parâmetros, exibindo graficamente o resultado da simulação.

4 RESULTADOS E DISCUSSÕES

Desenvolvida a modelagem de simulação discreta aplicada a comunicação entre uma interface homem-máquina, a validação da ferramenta se faz necessária utilizando de exemplo fundamentado.

Para que a interação dentro do escopo da simulação ocorresse, foi necessário a elaboração de duas rotinas a serem gerenciadas pelo recurso “Gerir estoque”.

Dessa forma, foi desenvolvido um fluxograma para cada uma das rotinas executadas pelo recurso “**Gerir estoque**”. A primeira rotina trata-se da solicitação de consumo do depósito, solicitação está que interage com a entidade estoque, avaliando se é possível ou não atender a demanda gerada pelo recurso, conforme Figura 09.

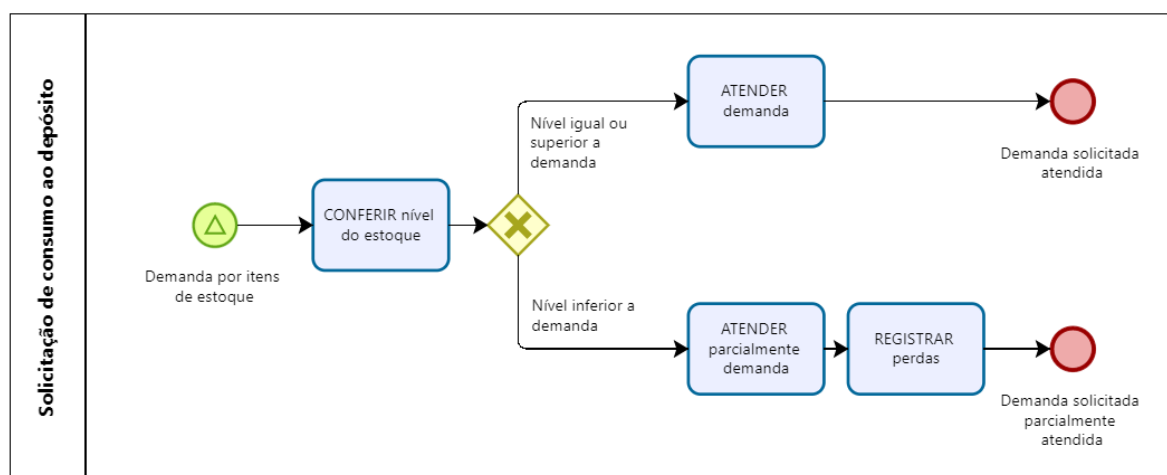


Figura 09. Solicitação de consumo do depósito (Fonte: autoria própria).

A segunda rotina trata da solicitação de ressuprimento periódica, acionada em intervalos regulares definidos na simulação, diretamente pelo usuário, através da interface gráfica. O tamanho do lote é calculado de acordo com a diferença entre o estoque máximo e o nível de

estoque no momento da verificação, conforme determina o modelo de revisão periódica de estoques. O fluxo dessa rotina pode ser conferido na Figura 10.

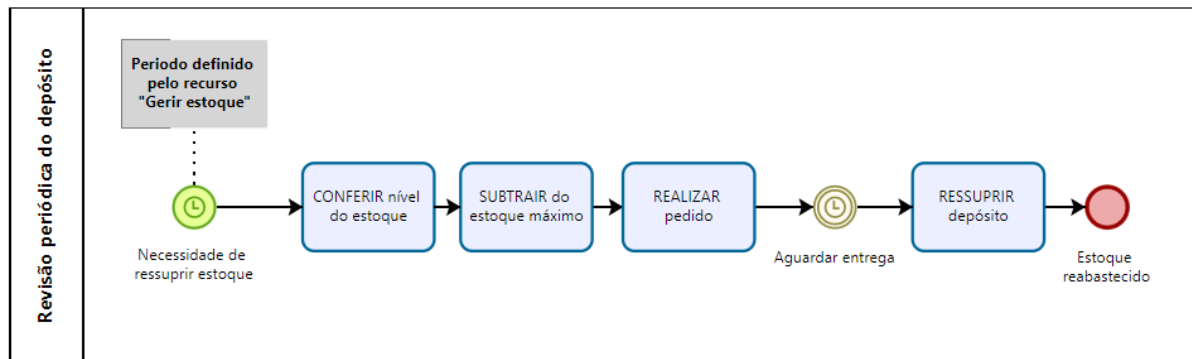


Figura 10. Momento de solicitar ressuprimento do estoque. (Fonte: autoria própria)

A implementação do gráfico foi feita utilizando uma biblioteca do Javascript destinada especificamente a criação de gráficos a partir de uma fonte de dados, chamada Chart.js. A biblioteca recupera os dados do servidor http através de uma rota chamada “./simulação”, contida no script do *Flask*, e a partir dela lê um arquivo JSON onde encontra-se todos os dados referentes ao nível de estoque. O arquivo JSON é um protocolo de comunicação entre servidores e clientes e é semelhante a um array, mudando apenas sua sintaxe, podendo ser multidimensional ou unidimensional. O array de nível de estoque servido pelo *Flask* é continuamente atualizado sempre que uma etapa da simulação é concluída, conforme pode ser visto na figura 11.

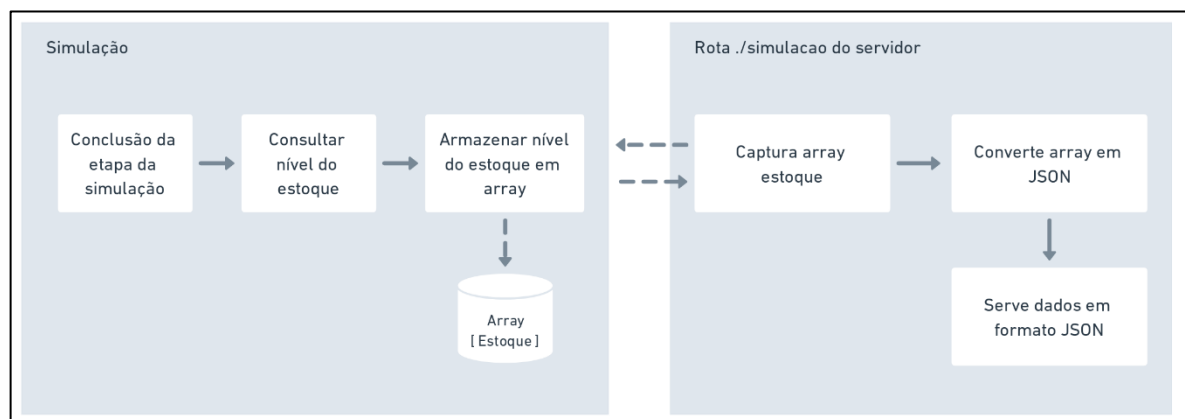


Figura 11. Acesso ao nível de estoque via JSON em servidor HTTP (Autoria própria).

Em artigo de Possamai *et al.* (2013), é apresentado estudo relacionado a política de controle de estoques, mais especificamente voltado a custos de estoque. Entretanto dentro da

problemática apresentada é desenvolvido o cálculo das variáveis necessárias a um modelo de revisão periódica, necessário para quantificar os custos de estoque.

Ao aplicar as variáveis de um dos produtos do estudo, o açúcar em saco de 50kg, foi possível validar as variáveis do estudo graficamente. Durante o período 365 dias foi executada a modelagem proposta no artigo e é possível identificar através do modelo computacional, sem a necessidade de formulas complexas ou grande conhecimento em softwares de simulação, que os atributos definidos no estudo são satisfatórios no que tangem ao nível de estoque mínimo, tamanho do estoque alvo e período de revisão, atributos esses que são condicionados a variáveis globais como demanda e tempo de entrega, além de custos que não cabem a esse estudo.

Com uma demanda de 418kg por dia, desvio padrão de 13kg diário, estoque máximo de 7524kg, período de revisão de estoque de 13 dias e tempo de ressuprimento de 5 dias tivemos a visualização da figura 12.

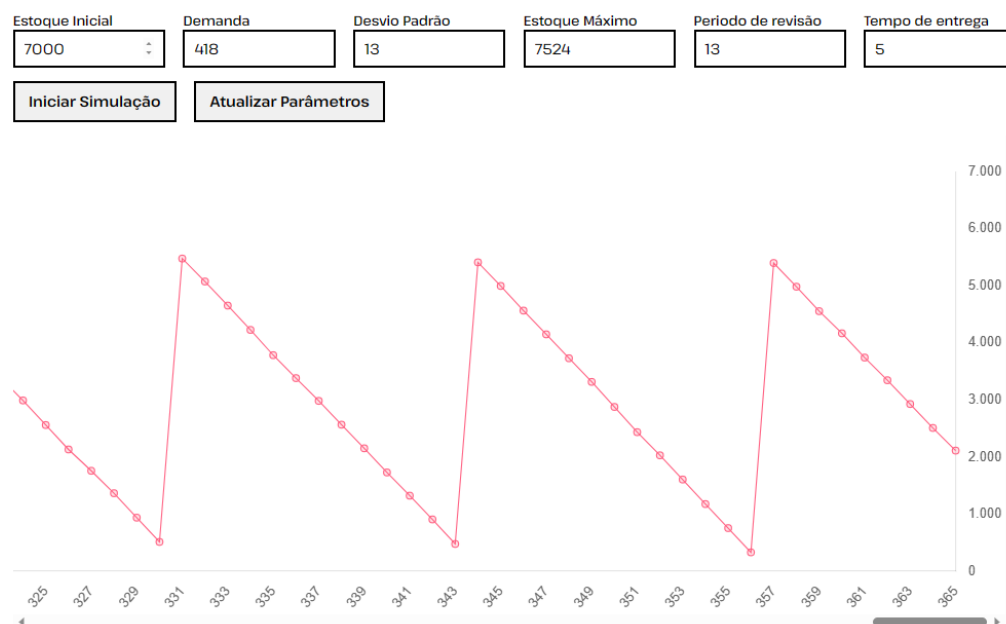


Figura 12. Validação do modelo com exemplo aplicado (Autoria própria).

A possibilidade de gerar as informações através de interface virtual, sem necessidade do uso de formulas ou conhecimento prévio em simulação, corrobora proposta de Nielsen, a interface conversa com o usuário pois os elementos contidos na mesma são comuns ao gestor de estoque, tornando a simulação uma ferramenta para atingir um meio.

5 CONSIDERAÇÕES FINAIS

A gestão de estoques é campo importante de estudo tanto na área acadêmica relacionada a gestão, que é pilar da inovação e evolução nas organizações, quanto na área empresarial, que precisa rotineiramente adequar-se a novas realidades geopolíticas, logísticas e de custo. A possibilidade de simular e visualizar o comportamento de dados contidos dentro do campo teórico ou mesmo da base de dados da organização, avaliando mudanças sutis ou consideráveis pode ser de grande valia.

Os softwares de simulação, entretanto, exigem do usuário conhecimento prévio do uso, a configuração de entidades, recursos e do próprio ambiente não só pode ser onerosa como também, a depender da complexidade do problema, pode exigir recursos que não estão disponíveis a organização. O desenvolvimento de tecnologia em simulação utilizando recursos como linguagens de programação e interfaces gráficas interativas abrem possibilidades na gestão de conhecimento das mais diversas áreas da engenharia e gestão.

Ao fim do estudo, a partir da validação do funcionamento da simulação discreta e possibilidade de interação com a mesma por meio de interface gráfica, é possível afirmar que os objetivos foram atingidos, pois a mesma demonstrou funcionamento pleno e total maleabilidade na manipulação de seus recursos e parâmetros por meio da interface gráfica projetada.

O modelo computacional de revisão periódica desenvolvido nesse estudo abre um leque de possibilidades aquele que possua interesse no estudo de gestão de estoques, desde validar um problema teórico disponível em livro-texto a validar uma situação real do dia-a-dia de um grande, médio ou pequeno negócio. A construção deste modelo computacional, aliado a possibilidade deste de iniciar ou alterar seus parâmetros via interface web, facilita o acesso a simulação discreta mesmo aqueles que não compreendem plenamente a aplicabilidade e funcionamento da mesma.

Os dados gerados a partir da simulação podem ser acessados visualmente, através de gráfico da interface web, ou de forma desestruturada, através de vetores gerados diretamente pelo SimPy e servidos através do Flask. É possível afirmar, portanto, que a ferramenta embora busque simplificar o acesso a simulação para gestão de estoques, pode ser utilizada para fins mais avançados e específicos. O objetivo principal, entretanto, a partir da validação do modelo computacional, foi cumprido.

Um dos pontos que merecem destaque no desenvolvimento do trabalho foi a dificuldade de integrar o backend da aplicação, que é o próprio modelo computacional, ao frontend, que é

a interface web, onde a interação é de fato realizada. Essa dificuldade se dá devido ao baixo volume de material acadêmico que aborde o tema de forma mais específica.

A mesma dificuldade, porém, dá ao trabalho um aspecto positivo que é o desenvolvimento de material na área que, por ventura, colabore para com aqueles que pretendam desenvolver soluções específicas a uma área específica do conhecimento no campo da simulação discreta.

5.1 Perspectivas de trabalhos futuros

Considerando as etapas do trabalho desenvolvido, sugestões para trabalhos futuros envolvem desde a aplicação do método desenvolvido em outras ferramentas de estoque como a revisão continua e modelos voltados a custo de estoque, até melhorias em relação ao modelo computacional e sua usabilidade, indo de encontro as 10 heurísticas de Nielsen. Outras opções de trabalhos futuros seriam a utilização do modelo computacional para fins específicos como a gamificação do ensino, adaptando a interface para fins educacionais. É desejável o avanço no tema e preferencialmente, que o modelo seja ampliado a fim de tornar-se uma ferramenta completa de simulação discreta orientada a gestão de estoques.

REFERÊNCIAS

- ANDRADE, M.** Introdução à metodologia do trabalho científico: elaboração de trabalhos na graduação, 10ª edição. Grupo GEN, 2012. E-book. ISBN 9788522478392. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788522478392/>. Acesso em: 26 abr. 2023.
- BEZERRA, M.** Gestão de estoques: definição de uma política de estoque para os itens mais críticos em uma empresa do setor elétrico da paraíba. Abepro, 2020. Disponível em: https://www.abepro.org.br/biblioteca/TN_STO_343_1758_40414.pdf. Acesso em: 04 mai. 2023.
- CHING, H.** Gestão de estoques na cadeia de logística integrada: Supply chain, 4ª edição. Grupo GEN, 2010. E-book. ISBN 9788522481293. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788522481293/>. Acesso em: 26 abr. 2023.
- CORDEIRO, R. et al.** Utilização de conceitos de interface homem-máquina para adaptação da disciplina de requisitos do RUP. 2009. Disponível em: <http://www.pos.cps.sp.gov.br/files/artigo/file/839/87b1ba6b0342dc35fa3e7c4a17db17de.pdf>. Acesso em 02 mai. 2023.
- CÔRREA, H.; GIANESI, I. G. N.; CAON, M.** Planejamento, programação e controle da produção: MRPII/ERP: conceitos, uso e implantação, 4. Ed. São Paulo: Atlas, 2001.
- DUONG, L.,** Discrete Event Simulation in Inventory Management. Auckland University. Disponível em: https://www.researchgate.net/publication/306092057_Discrete_Event_Simulation_in_Inventory_Management. Acesso em 29 abr. 2023
- FECOMERCIO,** quais ações adotar para tentar reduzir as perdas no varejo? Fecomercio, 2022. Disponível em: <https://www.fecomercio.com.br/noticia/adote-algumas-acoes-para-reduzir-as-perdas-no-varejo>. Acesso em: 26 abr. 2023.
- FUJIMAKI, R.** How AI Is Transforming Industries With Smart Inventory Management, SpiceWorks, 2023. Disponível em: <https://www.spiceworks.com/supplychain/warehousing/guest-article/ai-is-transforming-industries-with-smart-inventory-management/>. Acesso em 02 mai. 2023.
- GOBBO, V. F. et al.** Análise da adequação das heurísticas de usabilidade para avaliação de interfaces de equipamentos médicos. Abepro, 2017. Disponível em:

https://www.abepro.org.br/biblioteca/TN_STO_241_400_34499.pdf. Acesso em 05 mai. 2023.

HILLIER, Frederick S.; LIEBERMAN, Gerald J. Introdução à Pesquisa Operacional. Grupo A, 2013. E-book. ISBN 9788580551198. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788580551198/>. Acesso em: 26 fev. 2023.

HOLDEN, L. Inventory Optimization Using a SimPy Simulation Model. 2017. Disponível em: <https://dc.etsu.edu/etd/3219/>. Acesso em 05 mai. 2023.

INMAN, R., Enviromental Management: New challenges for production and inventory managers. Production and Inventory Management Journal. 1999. Disponível em: <https://www.proquest.com/openview/4bc0e437f8c317b2eea13d42fd11f96a/1?pq-origsite=gscholar&cbl=36911>. Acesso em: 29 abr. 2023.

JUNIOR, Murís L. Planejamento e Controle da Produção - Teoria e Prática. Grupo GEN, 2019. E-book. ISBN 9788521636304. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788521636304/>. Acesso em: 27 fev. 2023.

LAWRENCE, M., CHIROMO, F. (2020). An Analysis of the Inventory Management Challenges: The Case of Gypsum Plasterboard Manufacturing Company in South Africa.

MILLS ET AL. Using the Fetch API. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch. Acesso em: 03 mai. 2023.

NIELSEN, J. 10 Usability Heuristics for User Interface Design. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 6 mai. 2023.

NUMPY COMMUNITY. Numpy User Guide, 2022. Disponível em: <https://numpy.org/doc/1.23/numpy-user.pdf> . Acesso em: 04 mar. 2023.

PAOLESCHI, Bruno. ALMOXARIFADO E GESTÃO DE ESTOQUES. Editora Saraiva, 2019. E-book. ISBN 9788536532400. Disponível em: <https://app.minhabiblioteca.com.br/#/books/9788536532400/>. Acesso em: 27 fev. 2023.

PYTHON DOCUMENTATION. Disponível em: <https://docs.python.org/3/>. Acesso em: 03 mar. 2023.

POSSAMAI et al. Análise e seleção de uma política de controle de estoques com base em ferramentas e métodos da gestão de estoques. XXXIII encontro nacional de engenharia de

produção. Abepro, 2013. Disponível em: https://abepro.org.br/biblioteca/enegep2013_tn_stp_177_014_22262.pdf. Acesso em 06 mai. 2023.

SIMPY. **Read the Docs**. Disponível em: <https://simpy.readthedocs.io/en/latest/>. Acesso em: 03 mar. 2023.

SOUZA, T.; A Simulação a Eventos Discretos como ferramenta de apoio à tomada de decisão em empresas do ramo de mineração: aplicação em uma unidade da Yamana Gold
Disponível em:
https://www.repositorio.ufop.br/bitstream/123456789/3203/1/DISSERTA%C3%87%C3%83O_Simula%C3%A7%C3%A3oEventosDiscretos.PDF. Acesso em: 24 abr. 2023.

VAVRA et al.; Visualization in Science Education. ASEJ, vol. 41, p. 24 - 32, 2011.
Disponível em:
<https://sc.teachers.ab.ca/SiteCollectionDocuments/Vol.%2041,%20No.%201%20January%202011.pdf#page=24>. Acesso em: 09 de maio de 2023.

VERA et al.; Computer graphics applications in the education process of people with learning difficulties, ScienceDirect, 2007. Disponível em:
<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d8ea9dd8b409cb5905b846f8a878a15097bf9b67>. Acesso em: 09 de maio de 2023.

APENDICE A – MODELO COMPUTACIONAL DE SIMULAÇÃO

```
import simpy
import simpy.rt
import numpy as np
import threading
import json
from flask import Flask, Response, request
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

estoque_inicial = 0
pedido_lote = 0
tempo_entrega = 0
periodo_revisao = 0
nivel_maximo = 0
demanda_media = 0
desvio_padrao = 0
perdas = []
passagem = []
dias = 0
estoque = []
env = None

def startsimulation():
    class Deposito:
        global env
        global estoque
        dias = 0

    def __init__(self, env):
        self.env = env
        self.estoque = estoque_inicial
        self.action = env.process(self.gestao_estoque())

    def gestao_estoque(self):

        while True:
            if self.env.now != 0 and self.env.now % periodo_revisao == 0:
                quantidade_pedido = nivel_maximo - self.estoque
                if quantidade_pedido <= 0:
```

```

        quantidade_pedido = 0
        if quantidade_pedido > 0:
            self.env.process(self.fazer_pedido(quantidade_pedido))

        demanda = np.random.normal(demanda_media, desvio_padrao)
        yield self.env.timeout(1)
        if self.estoque < demanda:
            self.estoque -= self.estoque
            valorperdido = self.estoque - demanda
            perdas.append(round(valorperdido))
            self.dias += 1
        else:
            self.estoque -= demanda
            perdas.append(0)
            self.dias += 1

    def fazer_pedido(self, quantidade):
        yield self.env.timeout(tempo_entrega)
        self.estoque += quantidade

env = simpy.rt.RealtimeEnvironment(factor=1)
Deposito = Deposito(env)

def registrar_estoque(env, Deposito, estoque):
    while True:
        estoque.append(round(Deposito.estoque, 0))
        yield env.timeout(1)

env.process(registrar_estoque(env, Deposito, estoque))

try:
    env.run(until=float('inf'))
except KeyboardInterrupt:
    pass

def read_input():
    global demanda_media, desvio_padrao, dias_simulados

@app.route('/start')
def start():
    global estoque_inicial
    estoque_inicial = int(request.args.get('estoqueinicial'))

```

```

global demanda_media
demanda_media = int(request.args.get('demanda'))
global desvio_padrao
desvio_padrao = int(request.args.get('desvio'))
global nivel_maximo
nivel_maximo = int(request.args.get('pedido'))
global periodo_revisao
periodo_revisao = int(request.args.get('revisao'))
global tempo_entrega
tempo_entrega = int(request.args.get('entrega'))
startsimulation()
return f'ok'

@app.route('/simulacao')
def simulacao():
    jsondata = json.dumps(estoque)
    return Response(jsondata, status=200, mimetype='application/json')

@app.route('/atualizacao')
def parametros():
    global demanda_media
    demanda_media = int(request.args.get('demanda'))
    global desvio_padrao
    desvio_padrao = int(request.args.get('desvio'))
    global nivel_maximo
    nivel_maximo = int(request.args.get('pedido'))
    global periodo_revisao
    periodo_revisao = int(request.args.get('revisao'))
    global tempo_entrega
    tempo_entrega = int(request.args.get('entrega'))
    return f'ok'

if __name__ == '__main__':
    app.run()

while True:
    line = input()
    try:
        parts = line.split()
    except:
        print('Entrada inválida')

input_thread = threading.Thread(target=read_input)

```

```
input_thread.start()
```

APENDICE B – INTERFACE WEB

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<style>
  @import
url('https://fonts.googleapis.com/css2?family=Pathway+Extreme:wght@300;400;500;600;700
&display=swap');
  * {
    font-family: 'Pathway Extreme', sans-serif;
  }
  .inputs {
    display: flex;
    gap: 16px;
    font-family: sans-serif;
    align-items: flex-end;
    margin-bottom: 12px;
    box-sizing: border-box;
  }

  input {
    height: 30px;
    border: 2px solid black;
    max-width: 115px;
    padding: 0 8px;
  }
  .label {
    font-size: 12px;
    font-weight: 500;
  }

  button {
    border: 2px solid black;
    font-weight: 600;
    padding: 0 12px;
    height: 36px;
    font-size: 13px;
```

```
}

</style>
<body>
  <div class="inputs">
    <div class="content">
      <div class="label">
        Estoque Inicial
      </div>
      <input type="number" id="nmbInicial" name="nmbInicial">
    </div>
    <div class="content">
      <div class="label">
        Demanda
      </div>
      <input type="number" id="nmbDemanda" name="nmbDemanda">
    </div>
    <div class="content">
      <div class="label">
        Desvio Padrão
      </div>
      <input type="number" id="nmbDesvio" name="nmbDesvio">
    </div>
    <div class="content">
      <div class="label">
        Estoque Máximo
      </div>
      <input type="number" id="nmbPedido" name="nmbPedido">
    </div>
    <div class="content">
      <div class="label">
        Período de revisão
      </div>
      <input type="number" id="nmbRevisao" name="nmbRevisao" placeholder="Em
dias">
    </div>
    <div class="content">
      <div class="label">
        Tempo de entrega
      </div>
      <input type="number" id="nmbEntrega" name="nmbEntrega">
    </div>
  </div>
```

```
<div class="inputs">
  <button onclick="iniciarSimulacao()">Iniciar Simulação</button>
  <button onclick="atualizarParametros()">Atualizar Parâmetros</button>
```

```
</div>
<div class="chartcontent" style="width: 100%; overflow: scroll">
  <div class="chart-container" style="width: 100px">
    <canvas id="myChart" style="height: 70vh;"></canvas>
  </div>
</div>
```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
  let array = [];
  function PlotarGrafico() {
    fetch('http://127.0.0.1:5000/simulacao', { method: 'GET', mode: 'cors'})
      .then((response) => response.json())
      .then(data => {
        arrayEstoque = []
        for(let i = 0; i < data.length; i++) {
          arrayEstoque.push({x:i+1, y:data[i]})
        }

        var ctx = document.getElementById('myChart');

        var myChart = new Chart(ctx, {
          type: 'line',

          data: {
            labels: arrayEstoque.map(row => row.x),
            datasets: [{
              label: 'Estoque',
              backgroundColor: 'rgba(255, 99, 132, 0.2)',
              borderColor: 'rgba(255, 99, 132, 1)',
              borderWidth: 1,
              data: arrayEstoque.map(row => row.y)
            }],
          },
          options: {
            animation: false,
            maintainAspectRatio: false,
```

```

        scales: {
            y: {
                ticks: {
                    beginAtZero: true,
                },
                min: 0,
                grid: {
                    display: false
                },
                position: 'right'
            },
            x: {
                min: 0,
                grid: {
                    display: false
                },
            }
        }
    });
    function atualizarGrafico() {
        fetch('http://127.0.0.1:5000/simulacao', { method: 'GET', mode: 'cors'})
        .then((response) => response.json())
        .then(data => {
            tamanhoArray = data.length;
            arrayEstoque = []
            valor = 500
            if(tamanhoArray > 500) {
                for(let i = 0; i < 500; i++) {
                    arrayEstoque.push({x:tamanhoArray - valor, y:data[tamanhoArray -
valor]});

                    valor--;
                }
            } else {
                for(let i = 0; i < tamanhoArray; i++) {
                    arrayEstoque.push({x:i+1, y:data[i]})
                }
            }
            if(tamanhoArray > 500) {
                document.getElementsByClassName('chart-container')[0].style.width =
`$${80 + 500*20}px`;
            } else {

```

```

        document.getElementsByClassName('chart-container')[0].style.width =
        `${80 + tamanhoArray*20}px`;
    }

    document.getElementsByClassName('chartcontent')[0].scrollLeft =
document.getElementsByClassName('chartcontent')[0].scrollWidth
    myChart.data.labels = arrayEstoque.map(row => row.x);
    myChart.data.datasets[0].data = arrayEstoque.map(row => row.y);
    myChart.update();
    });
}

setInterval(atualizarGrafico, 1000);

});
}

PlotarGrafico()

function atualizarParametros() {

fetch(`http://127.0.0.1:5000/atualizacao?demanda=${document.getElementById('nmbDemanda').value}&desvio=${document.getElementById('nmbDesvio').value}&pedido=${document.getElementById('nmbPedido').value}&revisao=${document.getElementById('nmbRevisao').value}&entrega=${document.getElementById('nmbEntrega').value}`)
    }

function iniciarSimulacao() {

fetch(`http://127.0.0.1:5000/start?estoqueinicial=${document.getElementById('nmbInicial').value}&demanda=${document.getElementById('nmbDemanda').value}&desvio=${document.getElementById('nmbDesvio').value}&pedido=${document.getElementById('nmbPedido').value}&revisao=${document.getElementById('nmbRevisao').value}&entrega=${document.getElementById('nmbEntrega').value}`)
    PlotarGrafico()
}

</script>
</body>
</html>

```