

UNIVERSIDADE FEDERAL DE VIÇOSA

YARA CAROLINA DE SOUZA DA SILVA

**DESENVOLVIMENTO DE METODOLOGIA PARA CÁLCULO DE ESTOQUE EM
SILOS DE ARMAZENAGEM DE GRÃOS**

**VIÇOSA – MINAS GERAIS
2022**

YARA CAROLINA DE SOUZA DA SILVA

**DESENVOLVIMENTO DE METODOLOGIA PARA CÁLCULO DE ESTOQUE EM
SILOS DE ARMAZENAGEM DE GRÃOS**

Relatório final apresentado à Universidade Federal de Viçosa, como parte das exigências para obtenção do título de Bacharel em Engenharia Agrícola e Ambiental.

Orientador: Luis Cesar da Silva

Coorientadores: Augusto Cesar Laviola de Oliveira
Juliana Soares Zeymer

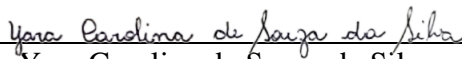
YARA CAROLINA DE SOUZA DA SILVA

**DESENVOLVIMENTO DE METODOLOGIA PARA CÁLCULO DE ESTOQUE EM
SILOS DE ARMAZENAGEM DE GRÃOS**

Relatório final, apresentado a Universidade Federal de Viçosa, como parte das exigências, para obtenção do título de Engenheira Agrícola e Ambiental.

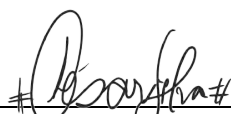
APROVADO: 24 de março de 2022.

Assentimento:



Yara Carolina de Souza da Silva

Autora



Luis Cesar da Silva

Orientador

AGRADECIMENTOS

À Deus, por estar presente em todos em todos os momentos da minha vida, me conduzindo sempre pelos melhores caminhos e fortalecendo minha fé diante das situações difíceis.

Ao meu pai, Toninho, por ser o meu maior exemplo de humildade e simplicidade. À minha mãe, Maria Helena, por todos os ensinamentos e por nunca ter deixado de acreditar em mim.

À minha irmã, Pâmela, pelo carinho e companheirismo e aos meus sobrinhos Davi, Elias e Lívia, por serem minha fonte de inspiração.

Ao professor Luís César, pela orientação, confiança, paciência e conhecimentos transmitidos. Aos coorientadores Juliana e Augusto, por terem sido tão solícitos na elaboração deste trabalho, me auxiliando muito, mesmo que de forma remota.

A todos os professores que contribuíram para que esta etapa tão importante em minha vida fosse concluída, de maneira especial a professora Fernanda.

À minha tia Ana, por sempre me incentivar e me ajudar em minhas decisões mais difíceis. Ao meu tio João Paulo, por ter sido um grande exemplo de fé e coragem. Às minhas primas Patrícia, Daniela e Débora e aos demais familiares e amigos, minha gratidão por me apoiarem sempre ao longo desta jornada.

À minha amiga Liz, por todo carinho e paciência, pelas infinitas horas de estudo, por ter compartilhado comigo os melhores sorrisos e também as lágrimas e por me ensinar o significado da amizade verdadeira. Ao meu amigo Edison por todo apoio e incentivo.

Aos meus amigos de curso Vinícius, Karen, Neri, Nathalia, Isabella, Dara, Camila, Thiago, Kerly e Angélica, pelos momentos de estudos e por momentos que fizeram com que os dias fossem mais leves e felizes. Às minhas amigas Maria e Cyndi parceiras e companheiras que contribuíram para que o meu estágio fosse o melhor do mundo.

À Universidade Federal de Viçosa, pela oportunidade de realizar o curso de Engenharia Agrícola e Ambiental durante esses anos e ao Departamento de Engenharia Agrícola (DEA), por ter fornecido toda estrutura necessária para o meu desenvolvimento.

À Procer, por ter fornecido os dados e todo suporte necessário para elaboração deste trabalho.

“Aquele que tem um porquê para viver pode enfrentar quase todos os como”.

(Friedrich Nietzsche)

RESUMO

SILVA, Yara Carolina de Souza da, Bacharel em Engenharia Agrícola e Ambiental, Universidade Federal de Viçosa, março de 2022. **Desenvolvimento de metodologia para cálculo de estoque em silos de armazenagem de grãos**. Orientador: Luís César da Silva. Coorientadores: Augusto Cesar Laviola de Oliveira e Juliana Soares Zeymer.

Em unidades armazenadoras de grãos, a estimativa das quantidades de grãos armazenados a granel é fundamental para quantificar os estoques e perdas. Atualmente, diferentes ferramentas para mensuração do estoque físico de grãos estão disponíveis, no entanto, são de alto custo, inviabilizando a aquisição por grande parte das unidades armazenadoras brasileiras. Dessa forma, objetivou-se com este trabalho a proposição de uma metodologia para o cálculo de estoque de arroz em casca armazenados em silos metálicos, empregando a linguagem de programação *Python*, métodos estatísticos, e dados da temperatura da massa de grãos medidos por meio de sensores digitais. Os dados de temperatura obtidos se referem a leituras realizadas diariamente às 06h00min no período entre 13 de abril e 16 de julho de 2021 por meio do sistema de gerenciamento Ceres[®]. A partir desses dados, foram calculadas as médias das temperaturas nas direções horizontal (por nível) e vertical (por pêndulo) e a geral. Em sequência, foi elaborada a matriz de ponderação, que teve os valores contrastados a média geral das temperaturas com o objetivo de mapear as alturas grãos em cada pêndulos. De posse dessas alturas, do fator de compactação e da massa específica do arroz em casca, foram calculados, para cada dia, o volume ocupado pela massa de grãos e estoque. O volume foi calculado em função do nível médio de produto nos pêndulos e por modelagem matemática. Os procedimentos foram desenvolvidos considerando arroz em casca, mas são perfeitamente aplicáveis a outros grãos. Conclui-se que, a tecnologia proposta possui relevância por facilitar o controle de estoque, dar suporte à tomada de decisão quanto ao manejo de estoque, condução da aeração de manutenção e minimização de perdas. E outro relevante benefício é a redução de risco de acidentes associados a rotinas de mensuração de estoque que envolve os acessos a alturas e ambientes confinados.

Palavras-chave: pós-colheita; estoque; tecnologia; *python*.

ABSTRACT

SILVA, Yara Carolina de Souza da, Bacharel em Engenharia Agrícola e Ambiental, Universidade Federal de Viçosa, March, 2022. **Development of a methodology for the calculation of stock in grain storage silos.** Adviser: Luís César da Silva. Co-advisers: Augusto Cesar Laviola de Oliveira and Juliana Soares Zeymer.

At the grain storage facilities, estimating the quantities of grain stored in bulk is essential to quantify inventories and losses. Nowadays, different tools for measuring the physical stock of grains are available, however, they are expensive, making the acquisition unfeasible by most Brazilian storage facilities. Therefore, the objective of this work was to propose a methodology for calculating the stock of paddy rice stored in metallic silos, using the Python programming language, statistical methods, and the grain mass temperature measured by means of digital sensors. Temperature data refer to readings taken daily at 6:00 am, from April 13 to July 16, 2021, through the Ceres® management system. Using these data, temperature averages were calculated in the horizontal (by level) and vertical (by pendulum) directions, and considering all sensors. In sequence, the weighting matrix was elaborated, which had the values contrasted with the general average temperature in order to map the grain heights in each pendulum. With these heights, the compaction factor, and the specific mass of paddy rice, the volume occupied by the mass of grains and stock was calculated for each day. The volume was calculated as a function of the average level of product in each pendulum and by mathematical modeling. The procedures were developed considering paddy rice, but are completely applicable to other grains. It was concluded that the proposed technology has relevance for facilitating stock control, supporting decision-making regarding stock management, conducting maintenance aeration, and minimizing losses. And another relevant benefit is the reduction of the risk of accidents associated with routines of inventory measurement that involve access to heights and confined environments.

Keywords: post-harvest; inventory; technology; python.

SUMÁRIO

1 INTRODUÇÃO	8
2 REVISÃO BIBLIOGRÁFICA	10
2.1 Produção e armazenagem de grãos no Brasil	10
2.2 Cubagem de silos	10
2.3 Propriedades físicas aplicadas na cubagem de silos	11
2.4 Tecnologias utilizadas na mensuração de estoque de grãos	12
2.5 Sistema de termometria	14
2.6 Linguagem de programação <i>Python</i>	14
3 MATERIAL E MÉTODOS	16
3.1 Caracterização do sistema e obtenção de dados	17
3.1.1 Distribuição espacial dos sensores e pêndulos	18
3.2 Desenvolvimento do código	20
3.2.2 Refinando a matriz de dados.....	21
3.2.3 Tratamento estatístico dos dados diários	21
3.2.4 Obtenção do estoque de grãos através do nível médio	22
3.2.5 Modelagem matemática	22
3.2.6 Obtenção do estoque de grãos a partir da modelagem matemática	24
3.3 Organização do algoritmo.....	25
4 RESULTADOS E DISCUSSÃO	28
4.1 Caracterização do sistema.....	28
4.2 Desenvolvimento do código	30
4.2.1 Distribuição espacial dos sensores e pêndulos	30
4.2.2 Refinando a matriz de dados.....	31
4.2.3 Tratamento estatístico dos dados	31
4.2.4 Obtenção do estoque de grãos através do nível médio	32
4.2.5 Modelagem matemática	32
4.2.6 Obtenção do estoque de grãos a partir da modelagem matemática	32
4.3 Funcionamento do algoritmo	33
5 CONCLUSÃO	38
REFERÊNCIAS	39
ANEXOS	42

1 INTRODUÇÃO

Em unidades armazenadoras de grãos, a mensuração do estoque físico é de extrema importância, sendo um dos primeiros passos para quantificar e controlar as perdas no armazenamento. Entretanto, elaborar o inventário de um produto agrícola a granel pode ser uma tarefa complexa e árdua. Para coleta de dados muitas vezes é necessário realizar trabalho em altura e acesso a espaços confinados que podem implicar em riscos de afogamento e, ou sufocamento na massa de grãos. A massa de grãos armazenada constitui-se de um meio poroso descontínuo formado pelos grãos que apresenta variação de suas propriedades físicas, tais como, dimensões, porosidade e massa específica. (THOMPSON et al., 1987, citados por ROCHA, 2020).

Geralmente, busca-se quantificar o estoque a partir da contabilização dos romaneios de recepção e expedição de produtos. No entanto, essa quantificação requer validação para conferência do estoque, o que no Brasil é feito por meio de cálculos de cubagem. A cubagem é uma operação que consiste em estimar a quantidade de grãos armazenados em função da massa específica, ângulo de repouso, fator de compactação do produto e o volume ocupado nas estruturas de armazenagem como os silos e os graneleiros (ROCHA, 2020).

Atualmente, diferentes ferramentas estão disponíveis para mensuração do estoque físico dos grãos, como: sensor ultrassônico, sensor do tipo radar, sensor a laser e sensor de nível pendular tipo Yo-Yo. Embora essas tecnologias se mostrem eficientes, geralmente apresentam custo elevado, se tornando um fator limitante para sua aquisição nas unidades armazenadoras de grãos. De acordo com uma consulta realizada em janeiro de 2022 o custo médio do sensor ultrassônico é de 5000 USD, da tecnologia do tipo 3D Radar é 6000 USD, do 3D Laser é 22000 USD e a tecnologia do tipo Yo-Yo é em torno de 2800 USD (EIP ENVIRO CONTROLS, 2022). Considerando-se o preço atual do dólar, percebe-se que tais equipamentos podem ser considerados onerosos ao sistema.

Com o intuito de promover a preservação da qualidade do produto durante o armazenamento, grande parte das unidades armazenadoras adotam sistema de aeração, em que um dos quesitos para a tomada de decisão quanto ao momento ideal para condução da operação é o monitoramento da temperatura da massa de grãos. Nesse monitoramento emprega-se sistemas de termometria que são compostos de cabos de aço para fixação, pêndulos e sensores de temperatura (SILVA, 2016). Outra aplicabilidade do sistema de termometria é a disponibilização de dados que podem ser utilizados em conjunto a equações matemáticas para determinar volume e a massa de grãos armazenada.

Dessa forma, objetivou-se com esse trabalho desenvolver uma metodologia para cubagem de silos armazenadores de grãos, a partir do emprego de recursos de programação, métodos estatísticos e a utilização de dados do sistema de termometria com sensores digitais de temperatura instalados em uma unidade armazenadora.

2 REVISÃO BIBLIOGRÁFICA

2.1 Produção e armazenagem de grãos no Brasil

Nos últimos cinquenta anos, o Brasil saiu da condição de importador para se tornar um grande provedor de alimentos para o mundo. Diante do contexto mundial e nacional são sinalizadas projeções de continuidade do crescimento da produtividade e da produção agrícola. Projeta-se para 2027, que o Brasil produza acima de 290 milhões de toneladas de grãos (EMBRAPA, 2018). Esses aumentos da produção agropecuária podem ser explicados pelos avanços na agricultura, com a utilização de pesquisas científicas e desenvolvimento tecnológicos, que visam o aumento da produtividade e a redução do tempo de produção, a partir do uso eficiente do solo, insumos, máquinas, implementos e tecnologias aplicadas ao cultivo (MACHADO JÚNIOR; REIS NETO, 2021).

Entretanto, a capacidade estática de armazenamento de grãos no país não tem acompanhado o aumento da produção, gerando anualmente o aumento do déficit da capacidade de armazenamento. As principais finalidades do armazenamento de grãos são a preservação da qualidade, minimização das perdas, e a agregação dos valores quanto a disponibilização do produto, matéria prima, ao longo do tempo e em local estrategicamente definido. No entanto, devido às carências quanto as disponibilidades de estruturas de armazenagem e de métodos e equipamentos para o monitoramento e controle de processos no armazenamento são potencializadas as perdas quantitativas e qualitativas (CORADI et al., 2020, citados por LUTZ; CORADI, 2022).

2.2 Cubagem de silos

A cubagem de silos e graneleiros consiste no emprego de métodos matemáticos para estimar o volume e a massa de produto armazenado. Nesse contexto, o estoque físico corresponde a quantidade real de produto existente na unidade armazenadora, que é aferido ou estimado por pesagem ou cubagem (COMPANHIA DE ENTREPÓSITOS E ARMAZÉNS GERAIS DE SÃO PAULO, 2013). Para a realização da cubagem de silos e graneleiros é necessário conhecer a massa específica do produto, o fator de compactação e as dimensões do espaço ocupado pela massa de grãos. A quantidade de produto estocada em estruturas de armazenagem pode ser calculada por meio da Equação 1 (SILVA, 2020):

$$Q_p = M_{esp} \times F_c \times V_p \quad (1)$$

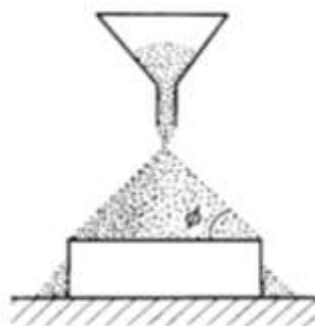
em que Q_p é a quantidade de produto (kg); M_{esp} , a massa específica do produto (kg/m^3); F_c , o fator de compactação (adimensional); e V_p , o volume ocupado pelo produto (m^3).

2.3 Propriedades físicas aplicadas na cubagem de silos

Para condução dos procedimentos de cubagem faz-se necessário o conhecimento das seguintes propriedades físicas dos grãos: massa específica, ângulo de repouso e fator de compactação. A massa específica corresponde a relação matemática entre a massa de um produto e o volume ocupado por ele. Portanto, para determinação da massa específica faz-se necessário ter um recipiente de volume conhecido, medir massa de produto acondicionada no recipiente e proceder o cálculo da relação matemática. A nível de laboratório podem ser empregados equipamentos como balança de peso hectolítrico e picnômetro. Além disso, alguns determinadores de umidade disponibilizam meios para determinação da massa específica.

O ângulo de repouso corresponde à inclinação do talude naturalmente formado pelo produto em relação ao plano horizontal e pode ser medido a partir do amontoado de produto granular ou pulverulento formado pelo seu basculamento sobre uma superfície plana (SILVA, CORRÊA, CALILI-JÚNIOR, GOMES, 2006). Sua obtenção pode ser feita através do escoamento do material a partir de um funil para uma superfície de acrílico, como pode ser observado na Figura 1.

Figura 1 – Método utilizado para medição de ângulo de repouso



Fonte: Juliano e Babosa-Cánovas citados por Campus 2012

Quanto ao fator de compactação, segundo BOTELHO et al. (2018) é observado que nos processos de transporte, pré-processamento e armazenamento, os grãos são frequentemente sujeitos à danos mecânicos e a compressão. Quando armazenados em silos, a massa específica

aumenta substancialmente devido às pressões de compressão ocasionadas pelo peso próprio do produto. Além disso, a massa específica pode ser influenciada por fatores como forma dos grãos, teor de água, dimensões, integridade física, condição sanitária, integridade biológica e a presença de impurezas (BOTELHO et al., 2018).

A compactação corresponde ao valor percentual da redução do volume de produto em razão da pressão da coluna de grãos (pressão de sobrecarga). Desse modo a compactação leva a alteração da massa específica em diferentes posições da massa de grãos acondicionadas em silos e graneleiros, Haque, E. (2013) verificou que o efeito da compactação é mais pronunciado na base do silo em função das pressões exercidas pelo peso próprio do material ensilado. Para se estimar o efeito da compactação nos cálculos da cubagem deve ser considerado o fator de compactação, Equação 2. Normalmente, esse fator de compactação varia entre 1 e 1,10 (SILVA, 2020).

$$Fc = \frac{100}{100 - comp} \quad (2)$$

Em que Fc é o fator de compactação; e $comp$ é a porcentagem de compactação.

2.4 Tecnologias utilizadas na mensuração de estoque de grãos

Atualmente, diferentes ferramentas estão disponíveis para mensuração do estoque físico de grãos armazenados, como: tecnologias de fotogrametria, sensor ultrassônico, sensor do tipo radar, sensor a laser e sensor de nível pendular tipo Yo-Yo.

A fotogrametria é a ciência que possibilita a obtenção de informação confiável a partir de processos de gravação, medição e interpretação de imagens fotográficas (ASP, 1966, citado por TEMBA, 2000). O uso da fotogrametria permite significativas melhoras na gestão de estoques e custos variáveis da cadeia produtiva, devido à facilidade no uso, versatilidade, agilidade e confiabilidade (MOGAI, 2021).

Já os sensores do tipo radar de ondas livres, a medição por ondas é transformada em dados que, por sua vez, são enviados para o quadro de comandos e disponibilizado para uma base de dados no portal com medição precisa em tempo real. Os dados fornecem informações do volume de produto e faz alterações instantâneas de movimentação de produto dentro do silo. A medição é feita mesmo em ambientes com poeira em suspensão (KEPLER WEBER, 2021).

Os sensores ultrassônicos tem uma aplicação em vários campos da engenharia, suas soluções são implementadas em dispositivos programáveis e, além disso, algoritmos podem ser incorporados para melhorar o desempenho da velocidade de processamento, precisão e detecção

da vazão. Estimar o sinal de eco do medidor ultrassônico é uma tarefa desafiadora, entretanto as tarefas de processamento para desempenho em tempo real são obrigatórias para melhorar a precisão nas medições (Senthil Kumar et al., 2020). As medições partem do princípio de que os impulsos ultrassônicos, gerados a partir de transdutores, viajam na velocidade do som e vêm refletidos pela superfície do meio, sendo enviados de volta ao transdutor. O período que decorre entre a emissão dos impulsos e a sua recepção dos mesmos pela antena é proporcional à distância existente entre a antena e o nível superior que se pretende medir. Dessa forma, o sensor ultrassônico transmite um pulso para a superfície do material, o pulso é refletido e recebido por um microfone e então o nível do material é medido (EIP ENVIRO CONTROLS, 2022).

O SPL-200, um tipo de sensor de nível a laser, mede sólidos em silos de até 30 metros de altura em ambientes com pouca ou nenhuma poeira. Realiza leituras diárias e horárias e é alimentado por uma bateria de lítio, que possui vida útil de 3 a 5 anos. O laser utiliza um transceptor integrado sem fio e as medições de dados são transmitidas para um aplicativo ou software (GRUNN, 2020).

O medidor pendular do tipo Yo-Yo tem o funcionamento baseado em um sistema eletromecânico composto por um pêndulo, cabo, motor e um circuito eletrônico. O eixo do motor fica preso a um carretel em que o cabo está enrolado. Na ponta deste cabo, encontra-se preso o pêndulo. A descida do pêndulo é gerenciada pelo motor e o circuito eletrônico mede o comprimento de cabo correspondente à distância do topo até o produto, por meio de um sistema ótico, convertendo o valor em um sinal analógico correspondente ao nível do material dentro do silo (NIVETEC, 2021). Ademais, existem sensores de nível de grão baseado em capacitância, que podem ser utilizados na observação contínua do conteúdo do silo, por meio de microcontroladores. Esse tipo de sensor não é afetado por fatores como umidade, temperatura e poeira dos grãos. Um circuito eletrônico é utilizado para processar os resultados da medição. A simulação do sensor demonstrou que as leituras são as previstas pela teoria desde que o sensor de referência esteja completamente preenchido com grãos, no entanto a leitura do sensor pode ser considerada falha, caso seja necessário saber a quantidade de grãos no silo através das leituras de nível devido ao acúmulo não plano de grãos no silo. Por outro lado, este tipo de erro é inevitável para qualquer sistema de medição de nível de sólidos, como grãos e pós (Işiker & Canbolat, 2009).

2.5 Sistema de termometria

A armazenagem de grãos necessita cada vez mais de equipamentos que sejam confiáveis e possam melhorar o rendimento e eficiência em cada área de atuação, como recebimento, classificação, secagem e armazenamento. A adoção de tecnologias que possibilitem melhorar resultados financeiros através de economia de energia e amenizar a perda de peso e qualidade do produto através da aeração é cada vez mais necessária (MONTEIRO, SOUZA, 2018). Além disso, existem equipamentos que permitem automatizar a gestão da operação de aeração. Os sistemas de aeração automatizados são constituídos de estação meteorológica, sistema de termometria e central computadorizada. A estação meteorológica possibilita a obtenção das condições psicrométricas do ar ambiente, o sistema de termometria propicia obtenção da temperatura da massa de grãos em diferentes pontos e a central computadorizada analisa os dados da termometria, as condições psicrométricas do ar ambiente, processa o programa e repassa informações ao sistema de acionamento dos ventiladores, de acordo com a opção selecionada pelo operador (SILVA, 2016).

Ainda de acordo com Silva (2016) o sistema de termometria é constituído de sensores de temperatura com espaçamento pré-determinado, conforme recomendado pela Instrução Normativa nº 29 de 2011, de no mínimo de um ponto de leitura a cada 150 m³ de grãos armazenados, que são conectados ao instrumento de leitura. A medição de temperatura pode ser feita através de sensores analógicos ou digitais. Na termometria analógica utiliza-se os termopares. Os termopares são constituídos de dois fios de metais diferentes unidos nas extremidades. Quando os pontos de união são expostos a temperaturas diferentes surge uma força eletromotriz (f.e.m.) que é dependente da temperatura. Os sensores digitais empregam a tecnologia “one wire” (um só cabo) e são construídos a base de metais semicondutores, estes sensores possuem três terminais: um para o envio de dado se os demais para alimentação elétrica.

2.6 Linguagem de programação *Python*

A primeira versão da linguagem Python foi lançada no ano de 1991 e desde então, a linguagem recebeu várias novas versões. A *Python Software Foundation (PSF)* é uma organização sem fins lucrativos que possui a propriedade intelectual relacionada à linguagem Python e é responsável pela curadoria e administração geral das versões da linguagem (DIVINO, 2022). Diferentemente do que é observado em outras linguagens, em que é muito

comum o uso excessivo de marcações, de marcadores e de palavras especiais, em *Python*, o uso desses recursos é reduzido, tornando a linguagem mais limpa e de fácil compreensão e leitura (FRANCO, 2016).

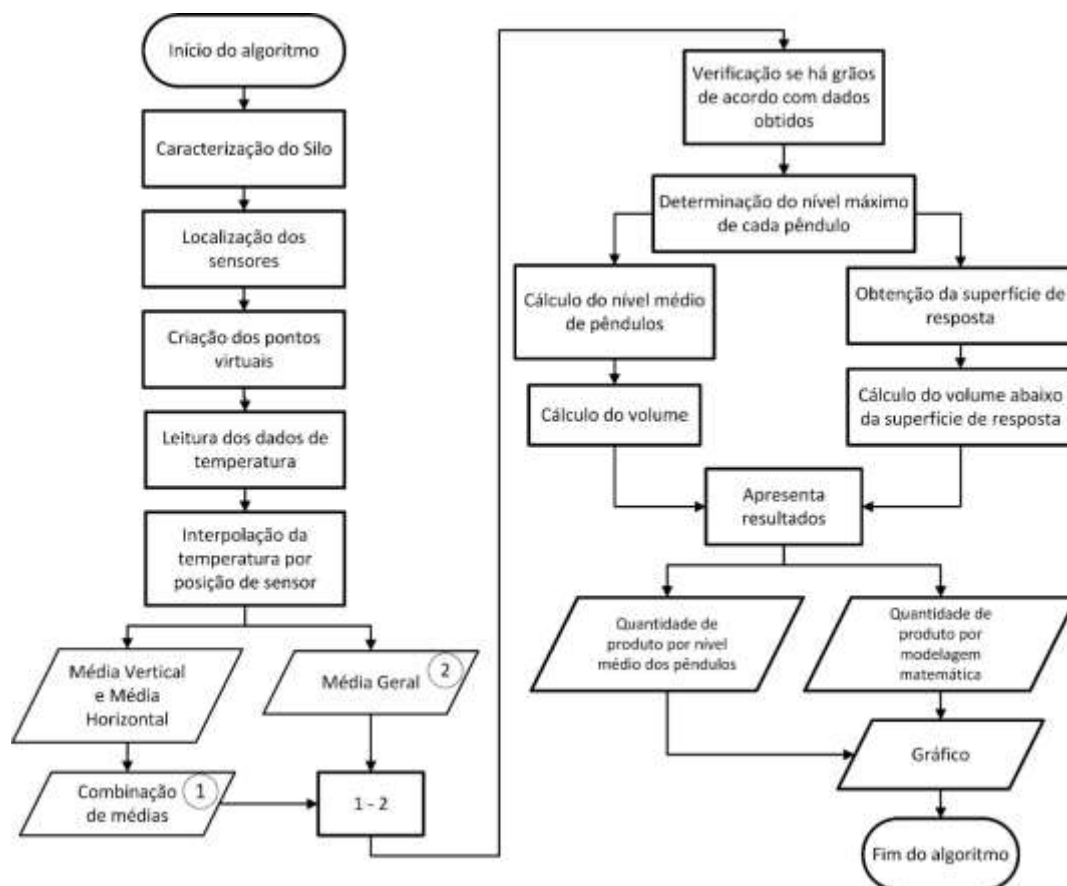
Python é uma linguagem de programação interpretada, ou seja, a implementação da linguagem no computador é feita através de um processo no qual um dos principais componentes é o interpretador, de alto nível, uma vez que possui maior proximidade com a linguagem humana do que com a linguagem de máquina e que suporta múltiplos paradigmas de programação como o imperativo, o orientado a objetos e o funcional (DIVINO, 2022).

Um grande número de desenvolvedores estão sempre elaborando bibliotecas, que podem ser importadas e executados. Quando são utilizadas bibliotecas como *NumPy*, que são responsáveis por operação de vetores e matrizes, *MatPlotLib*, referente a plotagem de gráficos e *SciPy* de ciência dos dados, matemática e engenharia, o *Python* passa a ser uma linguagem intermediadora de acesso às funções, o que reduz a quantidade de códigos e aumenta a produtividade. Ademais, o *Python* pode ser usado para o desenvolvimento programas para múltiplas plataformas e a partir dele é possível automatizar tarefas, o que pode contribuir de forma eficiente para o desenvolvimento de ideias, análise e processamento de dados (FRANCO, 2016).

3 MATERIAL E MÉTODOS

É apresentado na Figura 2 o fluxograma utilizado para o desenvolvimento do algoritmo para cubagem de silos. Inicialmente foi feita a caracterização do silo, em seguida foi definida a localização dos sensores de temperatura e foram criados pontos virtuais entre esses sensores de temperatura. Foi feita a interpolação da temperatura por posição de sensor. Após isso, foram obtidas a média geral de temperatura, a média vertical, a média horizontal e a combinação das médias vertical e horizontal. Foi feita a subtração entre a combinação das médias e a média geral e foi possível verificar os pontos do pêndulo onde existia grão. Após isso foi determinado o nível máximo de cada pêndulo e calculado o volume de grão de duas formas distintas, um a partir do nível médio dos pêndulos e outro a partir da modelagem matemática. Após isso o algoritmo apresenta os resultados, sendo fornecidas a quantidade de produto existente no silo de acordo com os dois volumes calculados e também foi gerado um gráfico mostrando a superfície de resposta. Dessa forma, o algoritmo foi finalizado.

Figura 2 – Fluxograma que descreve o algoritmo para cálculo da cubagem em silos armazenadores de grãos



Fonte: Autora.

3.1 Caracterização do sistema e obtenção de dados

O levantamento de dados para o desenvolvimento do algoritmo foi feito pelo sistema de gerenciamento Ceres[®] da empresa Procer Agroateligiência¹. Foram obtidos dados do sistema de termometria de um silo pertencente a uma unidade armazenadora localizada no estado de Santa Catarina. Esses dados referem ao perfil de temperatura na massa de grãos constatados no período de 13/04/2021 a 16/07/2021, ocorridos às 6:00 horas a cada dia. O silo, nesse período, estava acondicionado com arroz em casca, com teor de água igual a 15% (b.u.). Os parâmetros de entrada do sistema estão apresentados na Tabela 1.

Tabela 1 – Dados pré-cadastrados para execução do algoritmo

Parâmetro	Opção cadastrada
Tipo de grão	Arroz em casca
Compactação	5%
Diâmetro do silo	22,22 m
Fator de majoração da pressão estática	10 %
Fluxo de ar	1,5 m ³ /m ² .t
Ângulo de fundo	0°
Tipo de enchimento	Carga plano
Número de ventiladores	2
Rendimento dos ventiladores	77 %
Velocidade de rotação dos ventiladores	3600 rpm
Voltagem dos ventiladores	380 V
Corrente dos ventiladores	48,55 A
Potência dos ventiladores	40 cv
Número de queimadores	2
Voltagem dos queimadores	220,37 V
Corrente dos queimadores	0,2 A
Rendimento dos queimadores	90 %
Número de pêndulos	9
Número de sensores em cada pêndulo	16
Altura do primeiro sensor	0,7 m
Distância entre os sensores	1,6 m
Distância do pêndulo até a parede do silo	3,74 m

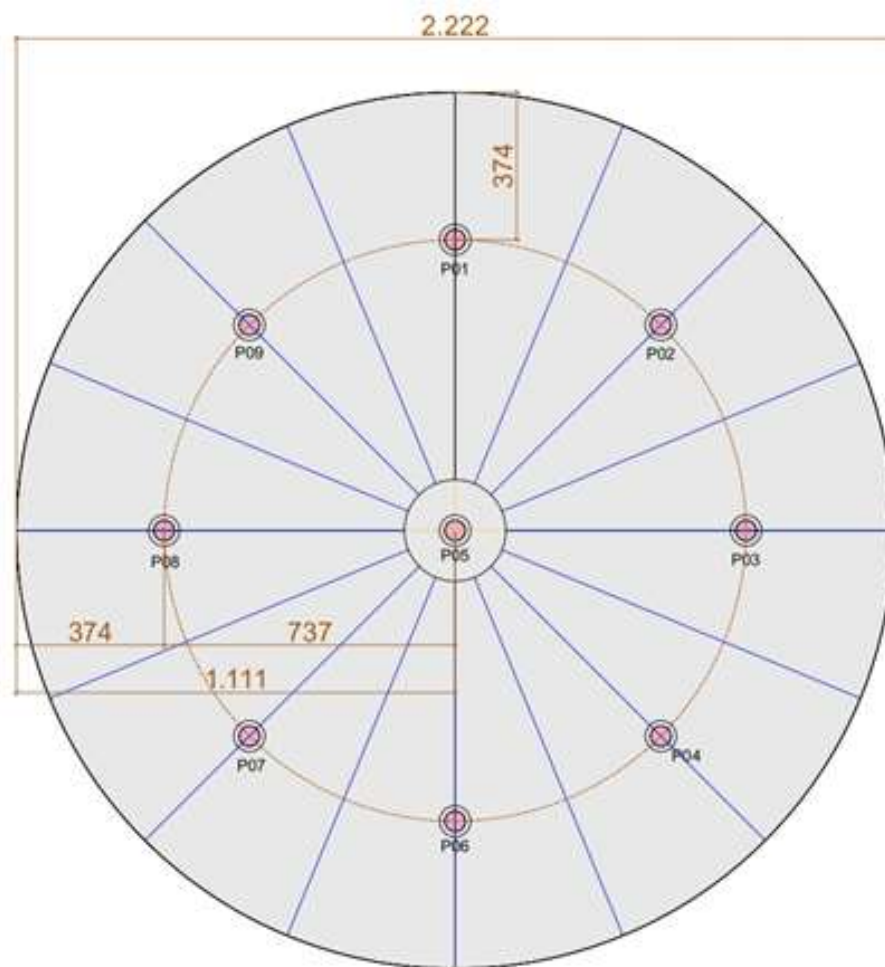
Fonte: Dados da pesquisa.

¹ Empresa brasileira, situada em Criciúma, Santa Catarina (www.procer.com.br).

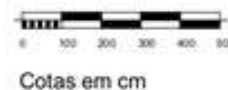
3.1.1 Distribuição espacial dos sensores e pêndulos

Na Figura 3 é apresentada a localização dos nove pêndulos do interior do silo, sendo um ao centro e os outros oito distribuídos radialmente de forma equidistante ao longo de uma circunferência com o raio de 7,37 m. O pêndulo central possui 16 sensores de temperatura e os demais, 14 sensores, sendo esses sensores espaçados em 160 cm.


Figura 3 – Posicionamento dos pêndulos no silo



Vista superior



LEGENDA:

-  Posição dos Pêndulos de Suporte dos Sensores de Temperatura Numerados de 01 a 09

Fonte: Autora.

3.2 Desenvolvimento do código

Na implementação do código foi empregada a linguagem de programação *Python* versão *Python 3.8* e foram utilizados recursos disponíveis nas bibliotecas *NumPy*, *SciPy*, *Matplotlib*. A Tabela 2 apresenta os tipos de variáveis que foram utilizadas no código.

Tabela 2 – Tipos de variáveis utilizadas no código

Variável	Tipo
Dss	numpy.ndarray
Ds	numpy.ndarray
D	numpy.ndarray
Dd	numpy.ndarray
med_geral	numpy.float64
med_hor	numpy.ndarray
med_vert	numpy.ndarray
dp1	numpy.ndarray
dp2	numpy.ndarray
dp3	numpy.ndarray
dp4	numpy.ndarray
dp5	numpy.float64
Teste1	lib_cub_pc.CapacEstatica
aux2	int
X1	numpy.ndarray
X2	numpy.ndarray
X3	numpy.ndarray
X4	numpy.ndarray
X5	numpy.ndarray
Y	sympy.core.symbol.Symbol
X	sympy.core.symbol.Symbol
Z	sympy.core.numbers.Float
dp6	float
Fig	matplotlib.figure.Figure
ax3d	matplotlib.axes._subplots.Axes3DSubplot
xdata	numpy.ndarray
ydata	numpy.ndarray
colors	numpy.ndarray
Z	numpy.ndarray
plt.show	function
Cont	int

Fonte: Autora

3.2.2 Refinando a matriz de dados

Conforme pode ser observado na Figura 3, por meio do projeto arquitetônico, o pêndulo central possuía 16 sensores de temperatura enquanto os demais 14. Desta forma a priori o conjunto de dados correspondentes a cada dia foram tratados como uma matriz 16 x 9. Como intuito de aprimorar a determinação do nível da massa de grãos em cada pêndulo em função dos valores de temperatura lidos foi empregado o recurso interpolação linear, em que no espaçamento entre os sensores de 1,60 m foi seccionado em dez partes. Desse modo, passou a se trabalhar com uma matriz 160 x 9.

3.2.3 Tratamento estatístico dos dados diários

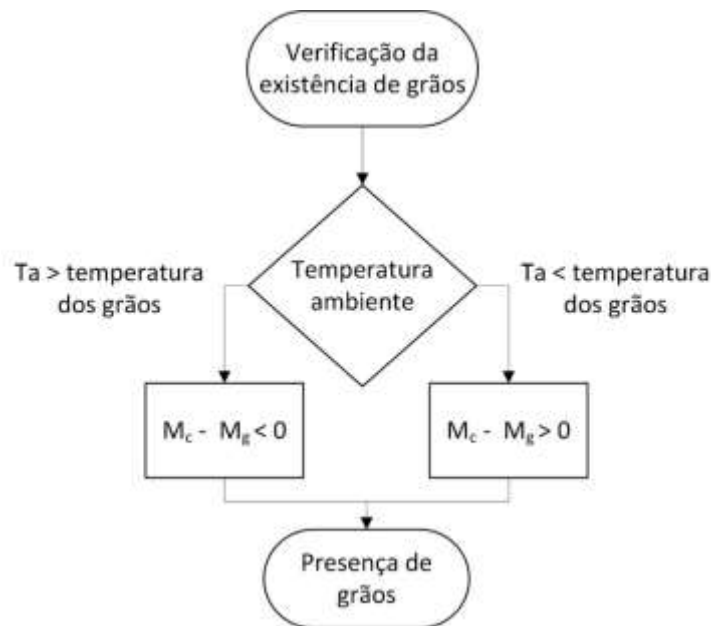
Utilizando-se a matriz de dados das leituras diárias, foram calculadas as médias nas direções horizontal (por nível) e verticais (por pêndulo) e a geral considerando todos os dados. Em sequência é calculada uma matriz de ponderação em que os elementos são calculados por meio da Equação 3.

$$T_{np} = \frac{(T_n \times N_s + T_p \times N_p)}{N_s + N_p} \quad (3)$$

Em que T_{np} é a temperatura ponderada no nível n e pêndulo p (°C); T_n , média de temperatura no nível n (°C); N_s , número de sensores no pêndulo central; T_p , a média de temperatura no pêndulo p (°C); e N_p , o número de pêndulos.

A partir dos valores de média de temperatura combinada, foi gerada uma nova matriz de dados, então, comparou-se a média combinada com a média geral de todos os sensores e, a partir dos valores obtidos, foi possível mapear os pontos onde havia grãos em cada posição dos pêndulos. A Figura 5 apresenta como foi feita a análise da existência de grãos em cada ponto.

Figura 5 – Verificação da existência de grãos



Fonte: Autora.

Para a situação em que a temperatura ambiente está mais elevada que a temperatura do grão, tem-se que quando a subtração entre a média combinada e a média geral resultar em um número negativo é indicativo de presença de grão, e quando o número obtido for positivo é indicativo de ausência de grão. Para a situação em que a temperatura do grão está mais elevada que a temperatura ambiente, tem-se que se a diferença entre a média combinada e a média geral resultar em um número positivo é indicativo de presença de grão e, se o número obtido for negativo, é indicativo de ausência de grão.

3.2.4 Obtenção do estoque de grãos através do nível médio

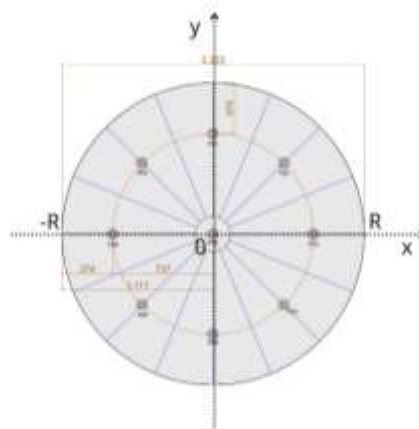
Para obtenção do volume através do nível médio, foi somado o nível de cada pêndulo e dividido pelo número de pêndulos. O fator de compactação foi obtido pela Equação 2, considerando 5% de compactação e massa específica considerada para o arroz em casca foi de 580 kg/m^3 . A Equação 1 foi inserida no código através de uma função denominada quantidade de produto e foi possível calcular a quantidade de arroz em casca armazenada no silo para cada dia.

3.2.5 Modelagem matemática

O plano cartesiano é utilizado para representar graficamente pares ordenados (x, y) de números reais. Identifica-se cada ponto de um plano com suas coordenadas em relação a um sistema com uma reta horizontal (eixo das abcissas) e outra reta vertical (eixo das ordenadas). O ponto de interseção (em ângulo reto) desses dois eixos é dito a origem do sistema. Cada ponto do plano corresponde, então, um par de coordenadas (x, y) , em que $|x|$ é a menor distância do ponto ao eixo das ordenadas, o que mostra que será uma reta que fará um ângulo de noventa graus com o eixo e $|y|$, a menor distância do ponto ao eixo das abcissas. A origem do plano cartesiano, denotada por O , tem, assim, ambas as coordenadas nulas (BEZERRA; COSTA E SILVA, 2010).

Para o silo em estudo, foi possível representar a posição de cada pêndulo e de cada sensor de temperatura em termos de pares ordenados (x, y) , considerando que o pêndulo central estava localizado na origem foi possível traçar a vista superior do silo no plano cartesiano, como pode ser observado na Figura 6.

Figura 6 – Vista superior do silo nos eixos x e y .



Fonte: Autora.

Analogamente ao que foi definido para plano o cartesiano, fixou-se três eixos mutuamente ortogonais e intersectando-os na origem, obteve-se o espaço cartesiano e dessa forma cada ponto no espaço cartesiano pôde ser representado a partir de coordenadas cartesianas x, y e z (BEZERRA; COSTA E SILVA, 2010).

Foi utilizada a metodologia de superfície de resposta, que consiste em uma coleção de técnicas estatísticas e matemáticas útil para o desenvolvimento, melhora e otimização de processos e pode ser aplicada na área industrial, em situações nas quais entram variáveis que podem influenciar na resposta de algum produto ou processo. Métodos computacionais vem

sendo desenvolvidos e podem ser utilizados para modelagem e análise de problemas a fim de obter grandes quantidades de dados de forma automática (COMPARINI et al., 2012).

Verifica-se que conhecendo o diâmetro do silo e a distância de cada pêndulo até a chapa lateral do silo é possível calcular as posições x e y para cada um dos pêndulos, e como eles estão espaçados a cada 45° , o cálculo pode ser feito por meio de algoritmos desenvolvidos no *Python*. Além das posições x e y , para descrever a altura de produto por pêndulos foi considerado o z . Dessa forma, obtém-se coordenadas cartesianas x , y e z e é possível elaborar o modelo matemático, Equação 4, que descreve uma superfície de resposta ao se ajustar os valores das constantes a , b , c , d , e , e f .

$$f(x, y) = f + eX + dY + cXY + bX^2 + aY^2 \quad (4)$$

Optou-se pela modelagem de uma superfície de segunda ordem devido a quantidade de dados disponíveis, o que limitaria o ajuste de um modelo mais refinado. A fim de verificar o desempenho da modelagem matemática foram calculados os coeficientes de determinação para cada modelo matemático.

3.2.6 Obtenção do estoque de grãos a partir da modelagem matemática

Um dos teoremas estabelecidos em cálculo propõe que para uma função $f(x,y)$ de duas variáveis, contínua numa região fechada R , no plano xy e com $f(x,y) \geq 0$ para todo (x,y) em R , o volume do sólido formado pelos pontos que estão abaixo do gráfico de $f(x,y)$ e tendo a região R como base, pode ser obtido pela integral apresentada na Equação 5 (LEITHOLD, 1994).

$$\iint_R f(x, y) dx dy \quad (5)$$

Uma circunferência de raio R , centrada na origem pode ser definida pela equação 6 (BEZERRA; COSTA E SILVA, 2010).

$$x^2 + y^2 = R^2 \quad (6)$$

Isolando o y , tem-se que $y = \pm\sqrt{R^2 - x^2}$.

Delimitando a integral na circunferência e integrando de $-R$ até R em x e de $-\sqrt{R^2 - x^2}$ até $\sqrt{R^2 - x^2}$ em y , tem-se:

$$\int_{-R}^R \int_{-\sqrt{R^2 - x^2}}^{\sqrt{R^2 - x^2}} f(x, y) dy dx \quad (7)$$

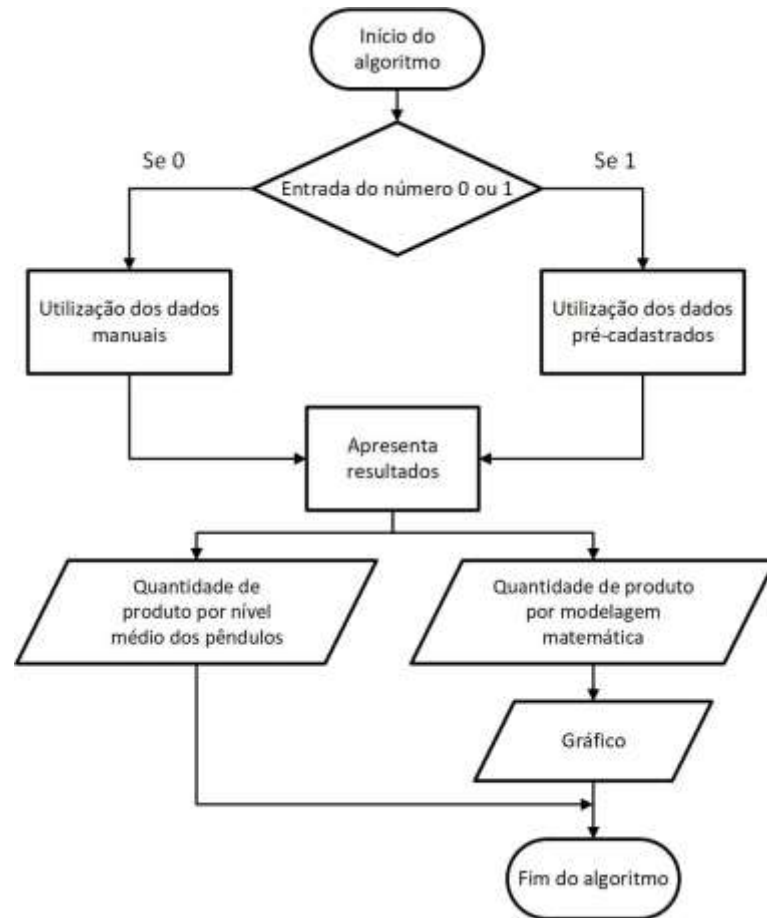
Considerando o teorema apresentado anteriormente pelas Equações 5, 6 e a definição de uma circunferência, foi possível obter o volume abaixo da superfície e acima do plano xy a partir da resolução da integral dupla da função de duas variáveis obtida anteriormente. Neste caso, o volume calculado corresponde ao volume de produto contido no silo.

Assim como no item 3.2.4 o fator de compactação foi obtido pela Equação 2, considerando 5% de compactação e considerando a massa específica para arroz em casca igual 580 kg/m^3 foi possível calcular a quantidade de produto utilizando a equação 1. No código o cálculo foi feito por meio da função denominada quantidade de produto.

3.3 Organização do algoritmo

Na Figura 7 é apresentado o fluxograma de funcionamento do programa. Ao executar o programa no *Python*, o usuário deverá digitar o valor 0, caso deseje informar todos os dados de entrada, ou 1, caso deseje que as informações sejam buscadas automaticamente. Assim que o programa lê os dados de entrada, os resultados são gerados pelo *console*, sendo fornecidos a quantidade de produto de acordo com o nível médio dos pêndulos e a quantidade de produto de acordo com a modelagem matemática e o gráfico para cada dia, mostrando o a superfície de resposta gerada pela modelagem matemática.

Figura 7 – Fluxograma de execução do programa



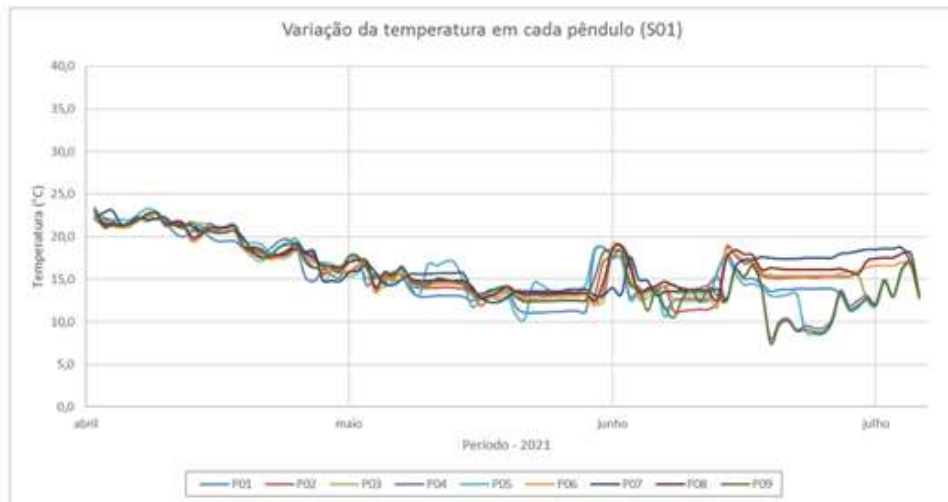
Fonte: Autora.

4 RESULTADOS E DISCUSSÃO

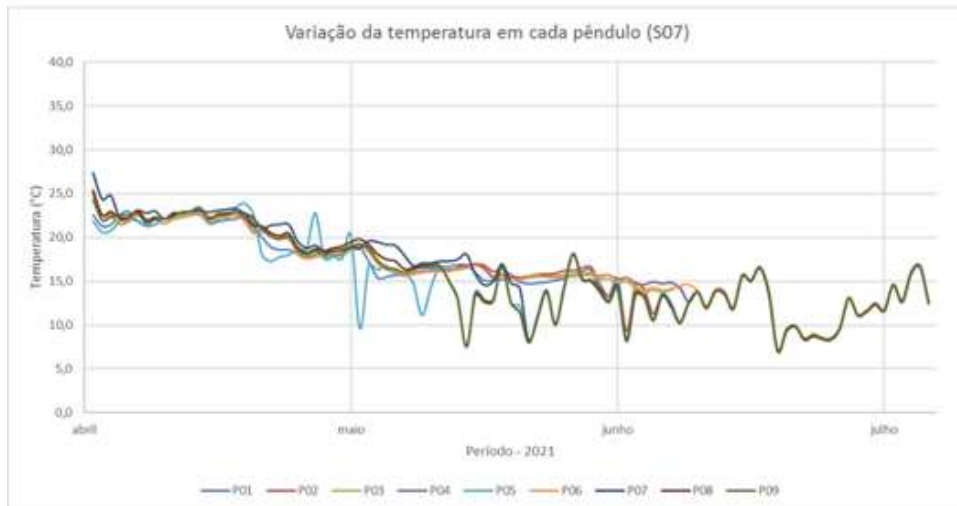
4.1 Caracterização do sistema

Na Figura 8 são apresentados gráficos que foram gerados a partir dos dados de termometria, correspondente ao período de 13/04/2021 a 16/07/2021 e referente a leituras realizadas às 6:00 horas. Na Figura 8a está apresentado um gráfico com as variações da temperatura por pêndulo para a posição dos sensores que estão localizados a 70 cm da base do silo, na Figura 8b está apresentado um gráfico com as variações da temperatura por pêndulo para a posição dos sensores que estão localizados na metade do silo e a Figura 8c apresenta um gráfico com as variações da temperatura por pêndulo para a posição dos sensores que estão localizados na parte mais elevada do silo.

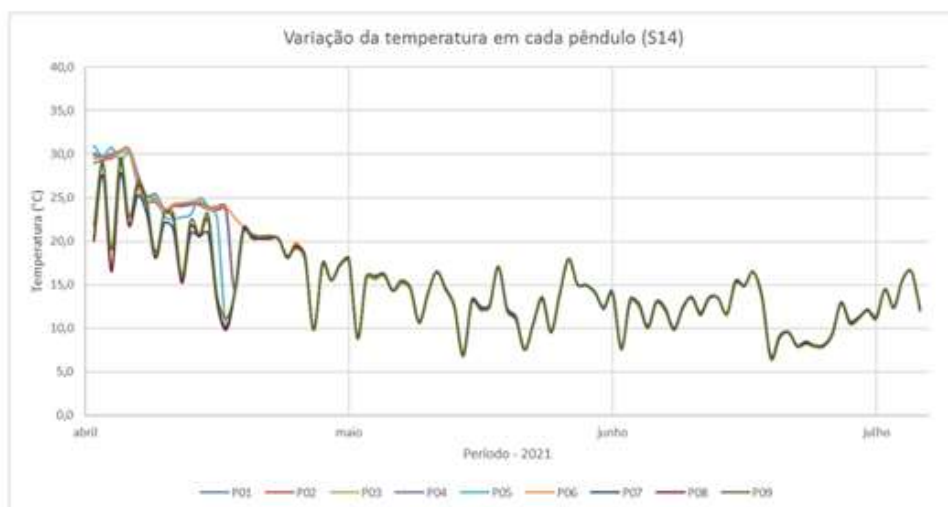
Figura 8 – Gráficos de variação da temperatura em cada pêndulo de acordo com a posição dos sensores



8a



8b



8c

Fonte: Autora.

Observa-se pela Figura 8c uma tendência de sobreposição das curvas a partir do final do mês de abril, o que é indicativo da presença de grãos. A temperatura da massa de grãos não apresenta grande variabilidade, mas a temperatura do espaço vazio, apresenta variabilidade associada a variabilidade da temperatura externa.

4.2 Desenvolvimento do código

O código desenvolvido em *Python* encontra-se disponível no Anexo I. Nas linhas de 9 até 24 do Anexo I são apresentadas as linhas iniciais do código, em que são relacionadas as bibliotecas inclusas ao código. Assim, por meio do comando *import*, os métodos disponibilizados pelas bibliotecas *NumPy*, *Matplotlib* e *SciPy* podem ser utilizadas no ambiente do programa desenvolvido.

Para interação com o usuário com vista a entrada de dados, utilizou-se a função *input()*, que faz com que o programa pare e espere o usuário digitar algum valor, e, ao teclar *enter*, o programa processa e imprime o valor digitado. Utilizou-se a estrutura de decisão definida pelo comando *if*, assim caso o usuário escolha a opção 0, será necessário informar todos os dados de entrada (tipo de grão, fator de compactação, diâmetro do silo, ângulo de fundo, tipo de enchimento, número de ventiladores, número de pêndulos, número de sensores, altura do primeiro sensor, distância entre os sensores, distância do pêndulo até a parede etc.). Caso o usuário escolha a opção 1 todas essas informações serão buscadas automaticamente, uma vez que já foram configuradas de acordo com os dados deste trabalho.

4.2.1 Distribuição espacial dos sensores e pêndulos

Conforme pode ser observado nas linhas de 114 até 123 do código implementado e apresentado no Anexo I criou-se uma matriz denominada *dss* para armazenar as posições dos sensores em relação a base do silo. O índice *j* da matriz representa os sensores que estão no mesmo nível e o índice *i* representa os pêndulos. Utilizou-se a estrutura *for*, uma vez que o número de pêndulos e sensores são conhecidos e sabe-se quantas vezes o código irá se repetir. A estrutura *for* é formada pelas variáveis *i* e *j*. A função *range()* gera uma lista com a localização vertical dos sensores. A estrutura irá fazer a variável *i* executar de 0 até o número de pêndulos e o *j* de 0 até o número de sensores. A cada iteração o valor da variável *i* e *j* assumem um valor dessa lista. Para *j=0*, que será o primeiro nível de sensores, os sensores irão receber altura 1 e

para os demais sensores ($j > 0$) irão receber o dado de altura abaixo dele mais a distância entre sensores. E assim por diante, até se obter a localização de todos os sensores. Assim, se por exemplo, na primeira entrada o i for 70 cm, na segunda será 70 cm somado a 160 cm e assim sucessivamente, até chegar ao último sensor.

4.2.2 Refinando a matriz de dados

Após definir a matriz de localização dos sensores foi necessário realizar a interpolação dos dados de temperatura, adicionando-se 10 pontos virtuais entre cada sensor de temperatura e entre os sensores localizados na posição inferior e a base do silo. As linhas de 127 até 134 apresentadas no Anexo I apresentam a parte do código em que se cria uma matriz (ds) com os valores dos pontos virtuais.

4.2.3 Tratamento estatístico dos dados

Nas linhas de 174 até 190 do código apresentado no Anexo I são exibidos os tratamentos estatísticos de dados. Calcula-se para os dados de temperatura a média geral, e as médias por nível e pêndulos. Faz-se a combinação das médias, multiplica-se por 10 uma vez que se considera a matriz refinada então é obtida a matriz $dp1$. Após essa etapa, subtrai-se a média combinada da média geral e obtém-se a matriz $dp2$. Realiza-se a verificação da existência de grãos e cria-se a matriz $dp4$ que é uma matriz de zeros e 1, que pode ser observada nas linhas de 192 até 212 do código. Assim, encontra-se o nível de grãos em cada pêndulo.

4.2.4 Obtenção do estoque de grãos através do nível médio

Para obtenção do volume através do nível médio foi necessário inserir no código a variável $dp5$, que foi obtida somando o nível de cada pêndulo e dividindo pelo número de pêndulos. Após isso, utiliza-se a função denominada capacidade estática que calcula a capacidade estática de silos. Esse processo está apresentado nas linhas de 214 até 263, apresentadas no Anexo I.

4.2.5 Modelagem matemática

As linhas de 265 até 299 do código exposto no Anexo I apresentam a inserção da modelagem matemática no código. É possível calcular a projeção x e a projeção y de cada pêndulo, com base na posição dos pêndulos e sensores. Ajusta-se um modelo de segunda ordem e obtém-se uma função que representa a superfície desejada.

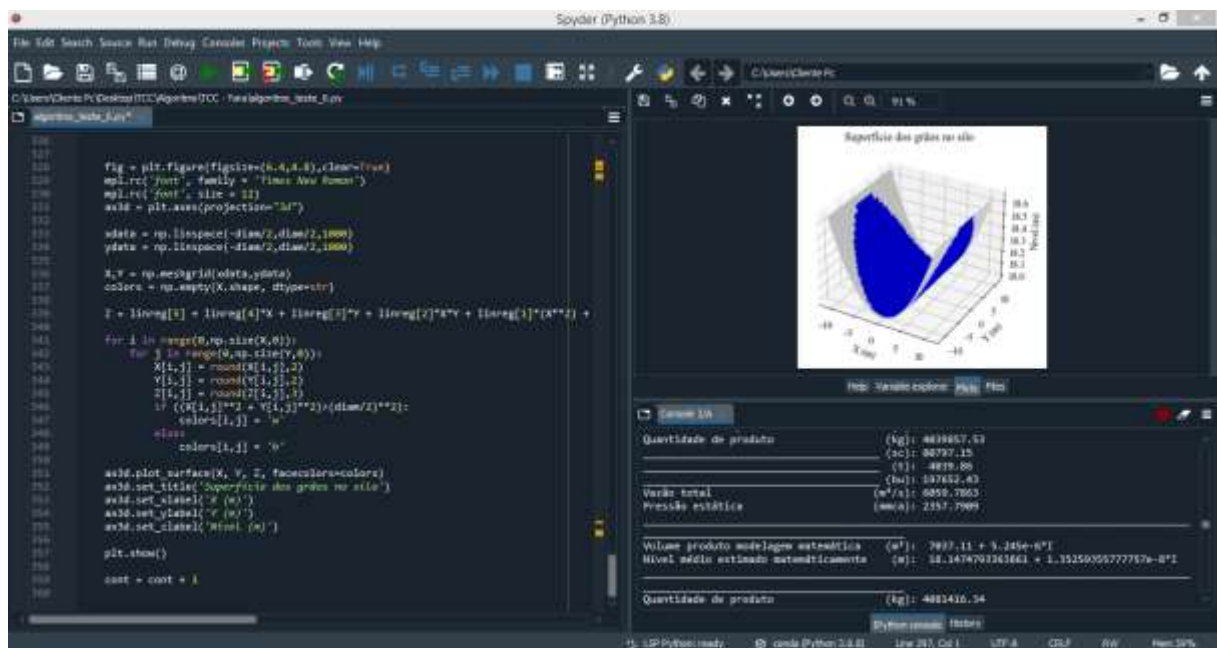
4.2.6 Obtenção do estoque de grãos a partir da modelagem matemática

As linhas de 301 até 360 do Anexo 1 apresentam a parte do código que possibilita a obtenção do volume abaixo da superfície e acima do plano xy a partir da resolução da integral dupla da função de duas variáveis obtida anteriormente. Este é o volume de produto que está sendo armazenado no silo. E a partir dele calcula-se o estoque de produto no silo. Por fim, as linhas de 362 até 393 presentes no Anexo I apresentam a parte do código que são responsáveis por gerar os gráficos de superfície de resposta.

4.3 Funcionamento do algoritmo

Ao escolher a opção executar o algoritmo no *Python*, irá aparecer a seguinte mensagem: “Digite 0 para dados manuais ou 1 para dados pré-cadastrados:”, o programa fica aguardando o usuário digitar algum dos valores. Dessa forma, ao escolher a opção 0, será necessário informar todos os dados de entrada, por outro lado, ao escolher a opção 1, todas essas informações serão buscadas automaticamente. Assim que o programa lê os dados de entrada, os resultados são gerados e podem ser observados pelo *console* do *Python*, sendo fornecidas a quantidade de produto (em quilogramas, toneladas, sacas e bushel) de acordo com o nível médio dos pêndulos e com a modelagem matemática, além do gráfico da superfície de resposta para cada dia e outras informações como vazão total e pressão estática. A Figura 9 apresenta os resultados impressos para um dia, após a execução do programa.

Figura 9 – Resultados impressos após execução do programa desenvolvido



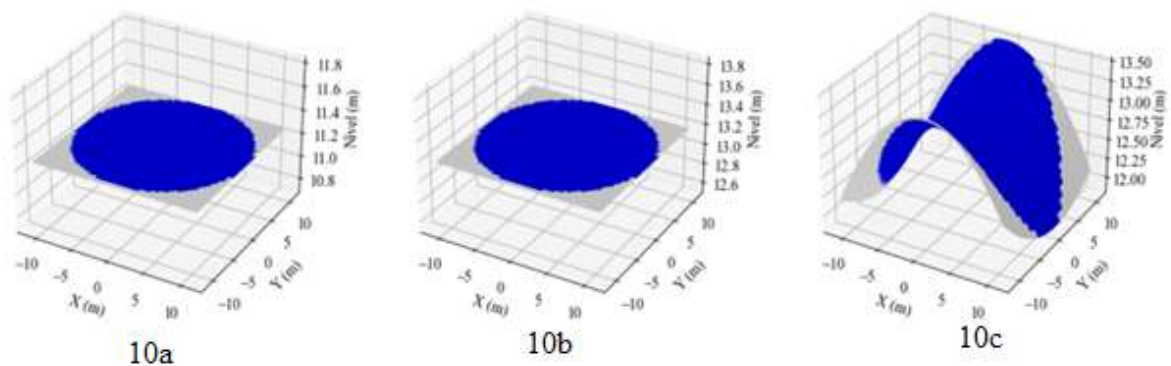
Fonte: Autora.

Ao executar o programa foi possível obter dois valores de estoque de arroz em casca para cada dia, um referente ao volume obtido através da média do nível dos pêndulos e outro referente ao volume obtido através da modelagem matemática.

De acordo com os resultados obtidos percebe-se que houve movimentação de produto na unidade armazenadora ao longo do período analisado. A Figura 10 apresenta os gráficos gerados pelo programa para os 3 dias iniciais da análise. A superfície formada pelos grãos está

representada em cada Figura na cor azul. Observa-se que da manhã do primeiro dia até a manhã do dia seguinte houve um aumento da quantidade de produto dentro do silo, o que indica que neste dia a unidade armazenadora estava recebendo produto. No terceiro dia foi verificada redução da quantidade de produto o que é indicativo de que houve mais expedições do que recepção de produto no dia anterior. As superfícies de resposta, indicada pelas Figuras 10a e 10b indicam que provavelmente houve aplainamento da camada de superior de grãos e a Figura 10c indica que houve descarga excêntrica. Os coeficientes de determinação obtidos para os dias 13 e 14 de abril, representados nas Figuras 10a e 10b, foram iguais a 1, uma vez que como a superfície observada foi plana, o seu volume coincidiu com o volume determinado a partir do nível médio dos pêndulos. Para o dia 15 de abril, que está representado pela Figura 10c, o coeficiente de determinação foi igual 0,0538, o que indica que o modelo de segunda ordem ajustado para esse dia apresentou baixo desempenho, podendo ser explicado pelo número limitado de dados para o cálculo das constantes de ajuste e pelo fato de não ter sido considerado o ângulo de repouso dos grãos na modelagem matemática.

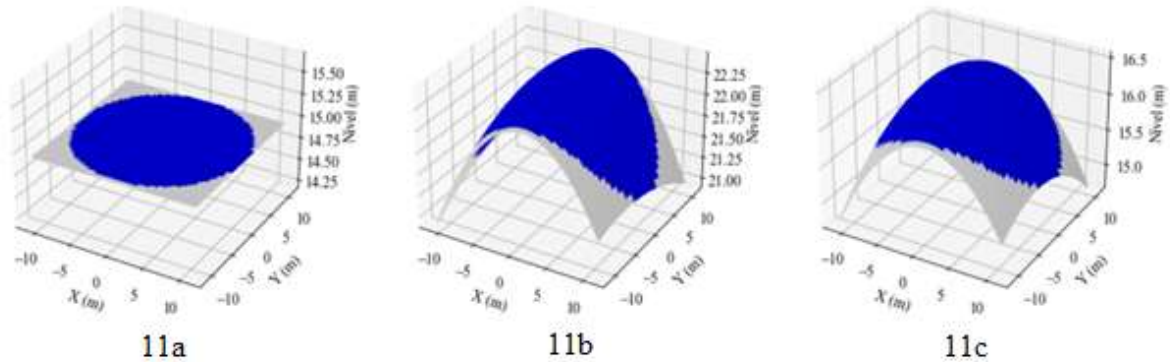
Figura 10 – Superfície dos grãos no silo no período de 13 a 15 de abril de 2021



Fonte: Autora.

No mês de maio, do dia 13 até o dia 15, observa-se que houve aumento da quantidade de produto dentro do silo, a Figura 11 retrata a disposição dos grãos, sugerindo que o silo recebeu produto neste período. Os coeficientes de determinação para os dias 13 de maio (Figura11a), 14 de maio (Figura11b) e 15 de maio (Figura11c), foram 1; 0,3092 e 0,2882, respectivamente.

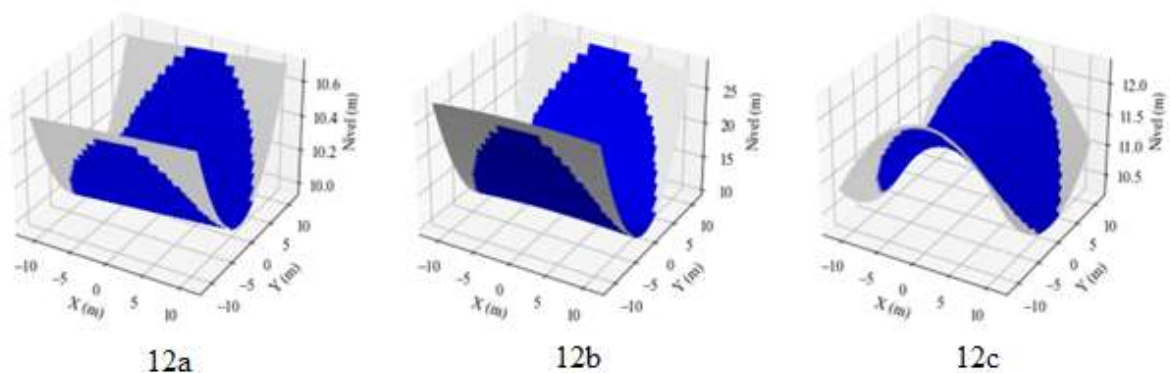
Figura 11 – Superfície de grãos no silo no período de 13 a 15 de maio de 2021



Fonte: Autora.

Em junho, nos dias 13, 14 e 15, como pode ser observado na Figura 12 observa-se redução da quantidade de produto dentro do silo, neste caso, as superfícies de resposta indicam que houve mais expedições do que recebimento de produto. Para o dia 13 de junho (Figura 12a) o coeficiente de determinação foi igual 0,2421, para o dia 14 de junho (Figura 12b) o coeficiente de determinação foi 0,033 e para o dia 15 de junho (Figura 12c) foi 0,1906. Novamente foram observados baixos valores de coeficientes de determinação, indicando que a modelagem necessita ser aprimorada.

Figura 12 – Superfície de grãos no silo no período de 13 a 15 de junho de 2021

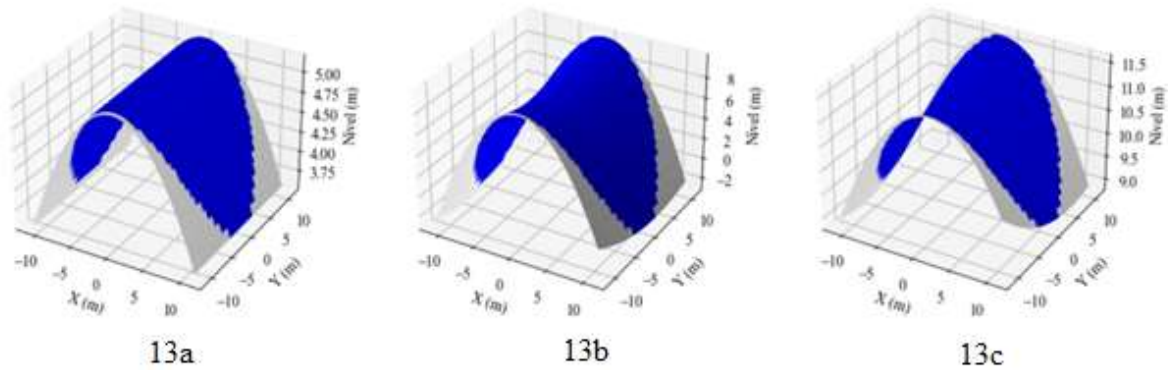


Fonte: Autora.

Finalmente para os últimos dias do período analisado, as superfícies de resposta, apresentadas na Figura 13, indicam que o silo ainda estava recebendo produto, o que é indicativo de que a unidade armazenadora continuava recebendo produto. Os coeficientes de

determinação para os dias 14 de julho (Figura13a), 15 de julho (Figura14b) e 16 de julho (Figura14c), foram 0,09202; 0,0001122 e 0,1769, respectivamente.

Figura 13 – Superfície de grãos no silo no período de 14 a 16 de julho de 2021



Fonte: Autora.

A Figura 14 apresenta a variação da quantidade de produto ao longo de todo o período analisado. A curva azul representa a quantidade de produto que foi obtida a partir da média do nível dos pêndulos e a curva magenta representa a quantidade de produto obtida pela modelagem matemática.

Figura 14 – Variação da quantidade de arroz em casca no silo de abril até julho de 2021



Fonte: Autora.

Comparando os resultados obtidos para todo o período de análise, observou-se que, em geral, as quantidades de produto obtidas a partir dos dois cálculos de volumes foram bem próximas, com exceção do dia 02/06/2021, em que o valor encontrado pela modelagem matemática foi muito superior à capacidade estática do silo. Este erro encontrado pode ser contornado pelo próprio programa, que faz a análise dos resultados, acusa o erro para o usuário e descarta o resultado que foi estimado de maneira incorreta. Percebe-se a importância da modelagem matemática criada, em relação ao cálculo grosseiro, uma vez que conceitos matemáticos podem ser utilizados para descrever a disposição de grãos dentro do silo.

5 CONCLUSÃO

A partir da utilização de dados de termometria e do emprego de recursos de programação e de métodos estatísticos foi possível desenvolver uma metodologia para cubagem de silos armazenadores de grãos. Por limitações de informações a respeito do estoque físico de grãos no silo ao longo do período analisado, não foi possível validar a metodologia proposta, entretanto esta validação poderá ser realizada futuramente.

A metodologia desenvolvida a partir de dados de arroz em casca pode ser aplicada para cálculo de estoque de outros grãos, e constitui em uma importante ferramenta de monitoramento em unidades armazenadoras. A tecnologia proposta é acessível aos armazenistas que já disponha de sistema de termometria, por ser de baixo custo de investimento e de grande utilidade, reduzindo riscos na realização de trabalho em altura e em espaço confinado, facilitando a rotina dos operadores e otimizando o controle de estoque.

Para trabalhos futuros, sugere-se o aperfeiçoamento da metodologia proposta de forma que na modelagem matemática seja considerado o ângulo de repouso dos grãos. Além disso, outra possibilidade seria a realização de uma interpolação considerando configurações geométricas e assim como foram obtidos sensores virtuais poderiam ser obtidos pêndulos virtuais. Sendo assim, haveria mais "pontos" para que pudessem ser desenvolvidos modelos mais refinados para as superfícies. Ademais, também poderia ser desenvolvida uma metodologia para o cálculo de estoque em silos, empregando sensores e dados das relativos à perda de carga do sistema de armazenagem.

REFERÊNCIAS

- BEZERRA, L. H.; COSTA E SILVA, I. P. **Geometria analítica**. 2.ed. Florianópolis: UFSC/EAD/CED/CFM, 2010. 170 p.
- BOTELHO, F. M. B.; FARIA, B. M. E. M.; BOTELHO, S. C. C.; RUFFATO, S.; MARTINS, R. N. Metodologias para determinação de massa específica de grãos. **Agrarian**, v. 11, n. 41, p. 251-259, 2018.
- CAMPOS, M. M. **Análise da escoabilidade de pós**. 2012. 129 f. Tese (Pós graduação em Engenharia Química) – Universidade Federal de São Carlos, 2012.
- COMPANHIA DE ENTREPÓS E ARMAZÉNS GERAIS DE SÃO PAULO. **Monitoramento de estoque físico de produtos armazenados**. São Paulo: Ceagesp, 2013. 19 p.
- COMPARINI, A.; PASSOS, G.; GRAZIADEI, H.; FERREIRA-SILVA, P. H.; LOUZADA, F. **Metodologia de superfície de resposta**: uma introdução nos softwares R e STATISTICA. São Carlos, SP: Instituto de Ciências Matemáticas e de Computação, 2012. 14 p. (Relatório Técnico do ICMC, 376).
- DIVINO, B. **Python – uma introdução à linguagem**. Disponível em: <https://www.alura.com.br/artigos/python-uma-introducao-a-linguagem>. Acesso em: 16 mar. 2022.
- EIP ENVIRO CONTROLS. **Level measurement solutions for over 3 decades**. Disponível em: www.eipenviroindia.com. Acesso em: 18 jan. 2022.
- EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA. **Visão 2030**: o futuro da agricultura brasileira. Brasília, DF: Embrapa, 2018. 212 p. Disponível em: <https://www.embrapa.br/visao/o-futuro-da-agricultura-brasileira>. Acesso em: 18 jan. 2022.
- FRANCO, J. L. **Introdução com a programação Python**. São Carlos: IFSP, 2016. Disponível em: http://antigo.scl.ifsp.edu.br/portal/arquivos/2016.05.04_Apostila_Python_-_PET_ADS_S%C3%A3o_Carlos.pdf. Acesso em: 03 mar. 2022.
- GRUNN. **Sensor de nível a laser com bateria**. São Paulo, SP: Grunn, 2020. Disponível em: <https://grunn.com.br/binmaster-medidores-de-nivel/sensores-a-laser-para-medicao-de-nivel/sensor-de-nivel-a-laser-com-bateria/>. Acesso em: 18 jan. 2022.
- Haque, E. (2013). **Estimating bulk density of compacted grains in storage bins and modifications of J ansen’s load equations as affected by bulk density**. Food Science & Nutrition, 1(2), 150–156. <https://doi.org/10.1002/fsn3.23>
- IŞIKER, H., & CANBOLAT, H. (2009). **Concept for a novel grain level measurement method in silos**. Computers and Electronics in Agriculture, 65(2), 258–267. <https://doi.org/10.1016/j.compag.2008.11.001>

KEPLER WEBER. **Sensor de nível radar**. Santos, SP: Kepler Weber, 2021. Disponível em: <https://youtu.be/dbERRpz7gUs>. Acesso em: 10 jan. 2022.

LEITHOLD, L. **O cálculo com geometria analítica**. 3.ed. São Paulo: Harbra, 1994.

LUTZ, E.; CORADI, P. C. Applications of new technologies for monitoring and predicting grains quality stored: sensors, internet of things, and artificial intelligence. **Measurement**, v. 188, 110609, 2022. <https://doi.org/10.1016/j.measurement.2021.110609>.

MACHADO JÚNIOR, P. C.; REIS NETO, S. A. **Perdas em transporte e armazenagem de grãos: panorama atual e perspectivas**. Brasília, DF: Conab, 2021. 197 p.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. **Instrução Normativa Nº29, de 8 de junho de 2011**. Lei do Sistema Nacional de Certificação de Unidades Armazenadoras, 2011. 14 p. Disponível em: <https://www.gov.br/agricultura/pt-br/assuntos/politica-agricola/infraestrutura-e-logistica/documentos-infraestrutura/29-2011.pdf>. Acesso em: 02 abr. 2022.

MOGAI. **Aumente a produtividade**: monitore os estoques de matérias-primas e produtos granulados. Vitória, ES: Mogai, 2021. Disponível em: <https://mogai.com.br/2021/10/27/aumente-a-produtividade-monitore-os-estoques-de-materias-primas-e-produtos-granulados/>. Acesso em: 18 jan. 2022.

MONTEIRO, E. F.; SOUZA, M. A. S. Termometria em unidades armazenadoras: comparativo de sensores digital e termopar. In: CONFERÊNCIA BRASILEIRA DE PÓS-COLHEITA, 7, 2018, Londrina. **Anais**. Londrina, ABRAMOS, 2018. p. 680-685.

NIVETEC. **Transmissor pendular Yo-Yo SiloPatrol SMU**. São Paulo, SP: Nivetec, 2021. Disponível em: <https://nivetec.com.br/produto/transmissor-pendular-yo-yo-silopatrol-smu/>. Acesso em: 18 jan. 2022.

ROCHA, J. C. **Compactação do arroz em casca no armazenamento a granel e seus efeitos sobre parâmetros de interesse operacional para a aeração**. 2020. 92 f. Tese (Doutorado em Ciência e Tecnologia de Alimentos) – Universidade Federal de Pelotas, 2020.

SENTHIL KUMAR, J., KAMARAJ, A., KALYANA SUNDARAM, C., SHOBANA, G., & KIRUBAKARAN, G. (2020). **A comprehensive review on accuracy in ultrasonic flow measurement using reconfigurable systems and deep learning approaches**. AIP Advances, 10(10). <https://doi.org/10.1063/5.0022154>

SILVA, F. S.; CORRÊA, P. C.; CALIL-JÚNIOR, C.; GOMES, F. C. Ângulo de repouso, atrito interno e efeito dos grãos de café com pergaminho. **Revista Brasileira de Produtos Agroindustriais**, v. 8, n. 1, p. 17-23, 2006.

SILVA, L. C. **Aeração de grãos: aplicabilidade e riscos**. Alegre, ES: UFES, 2016. Disponível em: https://www.agais.com/manuscript/ag0114_aeracao_aplicabilidade_risco.pdf. Acesso em: 19 jan. 2022.

SILVA, L. C. **Propriedades físicas dos grãos: amostragem, determinação de impurezas e umidade**. Viçosa: UFV/Centreinar, 2020. 31 p.

TEMBA, P. **Fundamentos da fotogrametria**. Belo Horizonte: UFMG, 2000. 24 p.

ANEXOS

Anexo I – Código desenvolvido para cálculo de estoque de grãos em silos armazenadores

```

1 """"*****
2 Finalidade: Desenvolvimento de um sistema para determinação de quantidade de produto
3 em silos armazenadores de grãos
4 *****
5 Última atualização: 01/04/2022
6 *****""""
7
8 #Declaração de Bibliotecas
9
10 from tkinter import *
11 from tkinter import messagebox
12 from numpy import *
13 from sympy import *
14 from sklearn.metrics import r2_score
15 import matplotlib.pyplot as plt
16 import matplotlib as mpl
17 from mpl_toolkits import mplot3d
18 import pandas as pd
19 import numpy as np
20 import lib_cub_pc as cpc
21 import math
22 import time
23 import sys
24 import os
25
26 #Setup do sistema
27
28 exemplo = int(input('Digite 0 para dados manuais ou 1 para dados pré-cadastrados: '))
29
30 if (exemplo==0):
31
32     linha= '_'*130
33     print(linha)
34     print('ENTRADAS DO SISTEMA')
35     print(linha)
36     print('Escolha o Tipo de Grão: \n 1 - Arroz com casca \n 2 - Café em coco \n 3 - Café
beneficiado \n 4 - Canola \n 5 - Girassol \n 6 - Milho espiga \n 7 - Milho granel')
37     print(' 8 - Soja \n 9 - Sorgo \n 10 - Trigo')
38     Tipo = int(input('Digite o número da opção: '))
39     Compactar = float(input('Fator de Compactação (%):'))
40     diam = float(input('Diâmetro do silo (m): '))
41     majorarPE = float(input('Fator de majoração da pressão estática (%): '))
42     fluxoT = float(input('Fluxo de ar em (m³/m².t): '))
43     angFundo = float(input('Ângulo de fundo (°): '))

```

```

44 print('Escolha o Tipo de Enchimento: \n 0 - Carga Plano \n 1 - Carga Cone \n 2 -
Descarga')
45 enchimento = int(input('Digite o número da opção: '))
46 nventilador = int(input('Número de ventiladores: '))
47 rventilador = float(input('Redimento dos ventiladores (%): '))
48 vrpm = float(input('Velocidade de rotação dos ventiladores (rpm): '))
49 voltagem_vent = float(input('Voltagem dos ventiladores (V): '))
50 corrente_vent = float(input('Corrente dos ventiladores (A): '))
51 pot_vent = float(input('Potência dos ventiladores (CV): '))
52 nqueimador = int(input('Número de queimadores: '))
53 voltagem_queim = float(input('Voltagem dos queimadores (V): '))
54 corrente_queim = float(input('Corrente dos queimadores (A): '))
55 rqueimador = float(input('Redimento dos queimadores (%): '))
56 num_p = int(input("Informe o número de pêndulos existentes na unidade armazenadora:
"))
57 num_s = int(input("Informe o número de sensores existentes em cada pêndulo: "))
58 alt1 = float(input("Informe a altura do primeiro sensor: "))
59 des = float(input("Informe a distância entre os sensores: "))
60 dist_par = float(input("Informe a distância do pêndulo até a parede: "))
61
62 if (exemplo==1):
63
64 linha= '_'*130
65 print(linha)
66 print('ENTRADAS DO SISTEMA')
67 print(linha)
68 Tipo = 1
69 print('Escolha o Tipo de Grão: \n 1 - Arroz com casca \n 2 - Café em coco \n 3 - Café
beneficiado \n 4 - Canola \n 5 - Girassol \n 6 - Milho espiga \n 7 - Milho granel')
70 print(' 8 - Soja \n 9 - Sorgo \n 10 - Trigo \n Tipo: ', Tipo)
71 Compactar = 5
72 print('Fator de Compactação (%): ', Compactar)
73 diam = 22.2200000001111
74 print('Diâmetro do silo (m): ', diam)
75 majorarPE = 10
76 print('Fator de majoração da pressão estática (%): ', majorarPE)
77 fluxoT = 1.5
78 print('Fluxo de ar em (m³/m².t): ', fluxoT)
79 angFundo = 0
80 print('Ângulo de fundo (°): ', angFundo)
81 enchimento = 0
82 print('Escolha o Tipo de Enchimento: \n 0 - Carga Plano \n 1 - Carga Cone \n 2 -
Descarga \n Opção: ', enchimento)
83 nventilador = 2
84 print('Número de ventiladores: ', nventilador)
85 rventilador = 77
86 print('Redimento dos ventiladores (%): ', rventilador)
87 vrpm = 3600
88 print('Velocidade de rotação dos ventiladores (rpm): ', vrpm)
89 voltagem_vent = 380

```

```

90 print('Voltagem dos ventiladores (V): ', voltagem_vent)
91 corrente_vent = 48.55
92 print('Corrente dos ventiladores (A): ', corrente_vent)
93 pot_vent = 40
94 print('Potência dos ventiladores (CV): ', pot_vent)
95 nqueimador = 2
96 print('Número de queimadores: ', nqueimador)
97 voltagem_queim = 220.37
98 print('Voltagem dos queimadores (V): ', voltagem_queim)
99 corrente_queim = 0.2
100 print('Corrente dos queimadores (A): ', corrente_queim)
101 rqueimador = 90
102 print('Redimento dos queimadores (%): ', rqueimador)
103 num_p = 9
104 print("Informe o número de pêndulos existentes na unidade armazenadora: ", num_p)
105 num_s = 16
106 print("Informe o número de sensores existentes em cada pêndulo: ", num_s)
107 alt1 = 0.7
108 print("Informe a altura do primeiro sensor: ", alt1)
109 des = 1.6
110 print("Informe a distância entre os sensores: ", des)
111 dist_par = 3.74
112 print("Informe a distância do pêndulo até a parede: ", dist_par)
113
114 dss = np.zeros((num_p,num_s))
115
116 for i in range(0,num_p,1):
117     for j in range(0,num_s,1):
118         if (j==0):
119             dss[i,j] = alt1
120         if (j>0):
121             dss[i,j] = dss[i,j-1]+des
122
123 #print(dss)
124
125 ds = np.zeros((num_p,num_s*10+1))
126
127 for i in range(0,num_p,1):
128     for j in range(0,num_s*10+1,1):
129         if (j<11 and j>0):
130             ds[i,j] = ds[i,j-1]+alt1/10
131         if (j>=11):
132             ds[i,j] = ds[i,j-1]+des/10
133
134 #print(ds)
135
136 #Leitura de dados do sensor
137
138 dados = pd.read_csv('dados_sensores_teste.csv', encoding = "ISO-8859-1", sep=';')
139 cont = 0

```

```

140 while(cont<np.size(dados,0)/10):
141
142   d = np.zeros((num_p,num_s))

143
144   for i in range(int(0+cont*10),int(num_p+cont*10),1):
145     for j in range(0,num_s,1):
146       d[int(i-cont*10),j] = dados.iloc[i,j]
147
148   for i in range(0+cont*10,num_p+cont*10,1):
149     for j in range(num_s-2,num_s,1):
150       d[int(i-cont*10),j]=dados.iloc[int(((num_p+1)/2-1)+cont*10),j]
151
152   dd = np.zeros((num_p,num_s*10+1))
153
154   for i in range(0,num_p,1):
155     aux = 0
156     for j in range(0,num_s,1):
157       if (j==0):
158         a = (d[i,j+1] - d[i,j])/(dss[i,j+1] - dss[i,j])
159         b = d[i,j+1] - a*dss[i,j+1]
160         for k in range(0,11,1):
161           dd[i,aux] = a*ds[i,aux] + b
162           aux = aux + 1
163       if (j>0):
164         a = (d[i,j] - d[i,j-1])/(dss[i,j] - dss[i,j-1])
165         b = d[i,j] - a*dss[i,j]
166         for k in range(0,10,1):
167           dd[i,aux] = a*ds[i,aux] + b
168           aux = aux + 1
169
170   #print(d)
171
172   #print(dd)
173
174   med_geral = sum(dd)/((num_s*10+1)*num_p)
175   med_hor = np.zeros((num_p,1))
176   med_vert = np.zeros((1,num_s*10+1))
177
178   for i in range(0,num_p,1):
179     med_hor[i,0] = sum(dd[i,:])/(num_s*10+1)
180
181   for i in range(0,num_s*10+1,1):
182     med_vert[0,i] = sum(dd[:,i])/(num_p)
183
184   dp1 = np.zeros((num_p,num_s*10+1))
185
186   for i in range(0,num_p,1):
187     for j in range(0,num_s*10+1,1):

```

```

188     dp1[i,j] = (med_hor[i,0]*num_p +
med_vert[0,j]*(num_s*10+1))/(num_p+num_s*10+1)
189
190     #print(dp1)
191
192     dp2 = dp1 - med_geral
193
194     verif = sum(dp2[0,num_s*10])
195
196     dp3 = np.zeros((num_p,num_s*10+1))
197
198     for i in range(0,num_p,1):
199         for j in range(0,num_s*10+1,1):
200             if (verif>=0):
201                 if (dp2[i,j]<=0):
202                     dp3[i,j]=ds[i,j]
203             if (verif<0):
204                 if (dp2[i,j]>=0):
205                     dp3[i,j]=ds[i,j]
206
207     dp4 = np.zeros((num_p,1))
208
209     for i in range(0,num_p,1):
210         dp4[i,0] = max(dp3[i,:])
211
212     #print(dp4)
213
214     dp5 = sum(dp4)/num_p
215
216     Teste1 = cpc.CapacEstatica(diam, dp5, Tipo, angFundo, enchimento, fluxoT,
nventilador, vrpm, rventilador, Compactar, majorarPE)
217
218     linha= ' '*130
219     # Gerar Relatório - Verificar e Validar _____
220     print(linha)
221     print('DIA: %8.0f' %(cont+1))
222     print(linha)
223     print('DADOS DO PRODUTO')
224     print(linha)
225     print('Produto                : %8.0f' %(Tipo))
226     print('Compactação                (%%): %8.2f' %(Compactar))
227     print('Massa específica do produto    (kg/m³): %8.2f' %(Teste1.mesp()))
228     print('Ângulo de Repouso do produto    (°): %8.2f' %(Teste1.ang_reposo()))
229     print(linha)
230     print('DADOS DO SILO')
231     print(linha)
232     print('Diâmetro                        (m): %8.2f' %(diam))
233     print('Altura média da coluna de grãos    (m): %8.2f' %(dp5))
234     print('Majoração da pressão estática      (%%): %8.2f' %(majorarPE))
235     print(linha)

```

```

236 print('DADOS DO VENTILADOR')
237 print(linha)
238 print('Número                : %8.0f' %(nventilador))
239 print('Voltagem                (V): %8.2f' %(voltage_vent))
240 print('Corrente de linha        (A)cpc.CapacEstatica(diam, dp5, Tipo, angFundo,
enchimento, fluxoT, nventilador, vrpm, rventilador, Compactar, majorarPE): %8.2f
%(corrente_vent))
241 print('Rendimento                (%%): %8.2f' %(rventilador))
242 print('Potência                (CV): %8.2f' %(pot_vent))
243 print(linha)
244 print('DADOS DO QUEIMADOR')
245 print(linha)
246 print('Número                : %8.0f' %(nqueimador))
247 print('Voltagem                (V): %8.2f' %(voltage_queim))
248 print('Corrente de linha        (A): %8.2f' %(corrente_queim))
249 print('Rendimento                (%%): %8.2f' %(rqueimador))
250 print(linha)
251 print('SAÍDAS DO BLOCO')
252 print(linha)
253 print('Área base                (m²): %8.2f' %(Teste1.area_base()))
254 print('Volume produto            (m³): %8.2f' %(Teste1.vol_produto()))
255 print('Nível médio de produto      (m): %8.2f' %(dp5))
256 print('Quantidade de produto (kg): %8.2f' %(Teste1.capKg()))
257 print('_____ (sc): %8.2f
%(Teste1.capSacas()))
258 print('_____ (t): %8.2f' %(Teste1.capTon()))
259 print('_____ (bu): %8.2f' %(Teste1.capBus()))
260 print('Vazão total                (m³/s): %8.4f' %(Teste1.vazao()))
261 print('Pressão estática            (mmca): %8.4f' %(Teste1.perdaCarga()))
262 print(linha)
263 # _____
264
265 aux2 = 0
266 posicao_pendulos = np.zeros((num_p,3))
267 for teta in range(0,360,45):
268     posicao_pendulos[aux2,0]=((diam/2)-dist_par)*np.cos(2*np.pi*teta/360)
269     posicao_pendulos[aux2,1]=((diam/2)-dist_par)*np.sin(2*np.pi*teta/360)
270     posicao_pendulos[aux2,2]=dp4[aux2,0]
271     aux2 = aux2 + 1
272
273 posicao_pendulos[8,0]=0
274 posicao_pendulos[8,1]=0
275 posicao_pendulos[8,2]=dp4[8,0]
276
277 X1 = np.zeros((1,num_p))
278 X2 = np.zeros((1,num_p))
279 X3 = np.zeros((1,num_p))
280 X4 = np.zeros((1,num_p))
281 X5 = np.zeros((1,num_p))
282 y = np.zeros((1,num_p))

```

```

283
284 for i in range(0,num_p,1):
285     X1[0,i] = posicao_pendulos[i,0]
286     X2[0,i] = posicao_pendulos[i,1]
287     X3[0,i] = X1[0,i]*X2[0,i]
288     X4[0,i] = X1[0,i]*X1[0,i]
289     X5[0,i] = X2[0,i]*X2[0,i]
290     y[0,i] = posicao_pendulos[i,2]
291
292 Z_real = y
293
294 y = np.transpose(posicao_pendulos[:,2])
295 X = [[X1[0,0],X1[0,1],X1[0,2],X1[0,3],X1[0,4],X1[0,5],X1[0,6],X1[0,7],X1[0,8]],
296      [X2[0,0],X2[0,1],X2[0,2],X2[0,3],X2[0,4],X2[0,5],X2[0,6],X2[0,7],X2[0,8]],
297      [X3[0,0],X3[0,1],X3[0,2],X3[0,3],X3[0,4],X3[0,5],X3[0,6],X3[0,7],X3[0,8]],
298      [X4[0,0],X4[0,1],X4[0,2],X4[0,3],X4[0,4],X4[0,5],X4[0,6],X4[0,7],X4[0,8]],
299      [X5[0,0],X5[0,1],X5[0,2],X5[0,3],X5[0,4],X5[0,5],X5[0,6],X5[0,7],X5[0,8]]]
300
301 X = np.transpose(X)
302 X = np.c_[X, np.ones(size(X,0))] # add bias term
303 linreg = np.linalg.lstsq(X, y, rcond=None)[0]
304
305 for i in range(0,6,1):
306     if ((linreg[i]<=0.000000000000000000000001) and (linreg[i]>=-
0.000000000000000000000001)):
307         linreg[i]=0.000000000000000000000001
308
309 Z_pred = linreg[5] + linreg[4]*X1 + linreg[3]*X2 + linreg[2]*X3 + linreg[1]*X4 +
linreg[0]*X5
310
311 verif_model_const = Z_real - sum(Z_real[0,:])/num_p
312
313
314 ccc = 0 # variável auxiliar contadora
315
316 for i in range(0,num_p):
317     if (verif_model_const[0,i] == 0):
318         ccc = ccc + 1
319
320 # se ccc for igual a num_p, o modelo é constante e adota-se o r2 como 1.
321
322
323 [Z_real] = Z_real
324 [Z_pred] = Z_pred
325 Z_real = list(Z_real)
326 Z_pred = list(Z_pred)
327
328 if (ccc == num_p):
329     r2 = 1.0000
330 else:

```

```

331     r2 = np.corrcoef(Z_real, Z_pred)[0,1]**2
332
333     print('Constante de ajuste do modelo      (f): ',linreg[5])
334     print('Constante de ajuste do modelo      (e): ',linreg[4])
335     print('Constante de ajuste do modelo      (d): ',linreg[3])
336     print('Constante de ajuste do modelo      (c): ',linreg[2])
337     print('Constante de ajuste do modelo      (b): ',linreg[1])
338     print('Constante de ajuste do modelo      (a): ',linreg[0])
339     print('Coeficiente de determinação do modelo (R²): ',r2)
340
341     init_printing(pretty_print=true)
342     x = Symbol('x')
343     y = Symbol('y')
344     z = Integral(linreg[5] + linreg[4]*x + linreg[3]*y + linreg[2]*x*y + linreg[1]*x**2 +
linreg[0]*y**2,(y,-(x**2+(diam/2)**2)**(1/2),(x**2+(diam/2)**2)**(1/2)),(x,-
(diam/2),(diam/2))).doit().evalf(6)
345     print('Volume produto modelagem matemática (m³): ',z)
346     dp6 = z/Teste1.area_base()
347     print('Nível médio estimado matematicamente (m): ',dp6)
348     print(linha)
349     z = round(z,3)
350     dp6 = round(z/Teste1.area_base(),3)
351     dp6 = float(dp6)
352
353     Teste2 = cpc.CapacEstatica(diam, dp6, Tipo, angFundo, enchimento, fluxoT,
nventilador, vrpm, rventilador, Compactar, majorarPE)
354     print('Quantidade de produto(kg): %8.2f' %(Teste2.capKg()))
355     print('_____ (sc): %8.2f'
%(Teste2.capSacas()))
356     print('_____ (t): %8.2f' %(Teste2.capTon()))
357     print('_____ (bu): %8.2f' %(Teste2.capBus()))
358     print('Vazão total          (m³/s): %4f' %(Teste2.vazao()))
359     print('Pressão estática          (mmca): %4f' %(Teste2.perdaCarga()))
360     print(linha)
361
362     fig = plt.figure(figsize=(6.4,4.8),clear=True)
363     mpl.rc('font', family = 'Times New Roman')
364     mpl.rc('font', size = 12)
365     ax3d = plt.axes(projection="3d")
366
367     xdata = np.linspace(-diam/2,diam/2,1000)
368     ydata = np.linspace(-diam/2,diam/2,1000)
369
370     X,Y = np.meshgrid(xdata,ydata)
371     colors = np.empty(X.shape, dtype=str)
372
373     Z = linreg[5] + linreg[4]*X + linreg[3]*Y + linreg[2]*X*Y + linreg[1]*(X**2) +
linreg[0]*(Y**2)
374
375     for i in range(0,np.size(X,0)):

```

```
376     for j in range(0,np.size(Y,0)):
377         X[i,j] = round(X[i,j],2)
378         Y[i,j] = round(Y[i,j],2)
379         Z[i,j] = round(Z[i,j],3)
380         if ((X[i,j]**2 + Y[i,j]**2)>(diam/2)**2):
381             colors[i,j] = 'w'
382         else:
383             colors[i,j] = 'b'
384
385     ax3d.plot_surface(X, Y, Z, facecolors=colors)
386     ax3d.set_title('Superfície dos grãos na unidade armazenadora')
387     ax3d.set_xlabel('X (m)')
388     ax3d.set_ylabel('Y (m)')
389     ax3d.set_zlabel('Nível (m)')
390
391     plt.show()
392
393     cont = cont + 1
```

Fonte: Autora.