

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIENCIAS TECNOLÓGICAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

VITOR ANTÔNIO ROMÃO

**INTELIGÊNCIA ARTIFICIAL PARA SIMULAÇÃO DE DISTORÇÕES DE
PEÇAS FUNDIDAS**

JOINVILLE

2025

VITOR ANTÔNIO ROMÃO

**INTELIGÊNCIA ARTIFICIAL PARA SIMULAÇÃO DE DISTORÇÕES DE
PEÇAS FUNDIDAS**

Proposta de dissertação submetida ao programa de Pós-Graduação em Engenharia Mecânica, PPGEM, do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como parte dos requisitos necessários para a obtenção do Título de Mestre.

Orientador: Marcus Vinícius Canhoto Alves

JOINVILLE

2025

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Universitária Udesc,
com os dados fornecidos pelo(a) autor(a)**

Romão, Vitor Antônio

Inteligência Artificial para simulação de distorções de
peças fundidas / Vitor Antônio Romão. -- 2025.

131 p.

Orientador: Marcus Vinicius Canhoto Alves

Dissertação (mestrado) -- Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas, Programa
de Pós-Graduação em Engenharia Mecânica, Joinville, 2025.

1. Inteligência artificial. 2. Fundição em areia. 3. Redes
neurais. 4. Simulação computacional. 5. Compensações
dimensionais. I. Alves, Marcus Vinicius Canhoto. II.
Universidade do Estado de Santa Catarina, Centro de
Ciências Tecnológicas, Programa de Pós-Graduação em
Engenharia Mecânica. III. Título.

VITOR ANTÔNIO ROMÃO

**INTELIGÊNCIA ARTIFICIAL PARA SIMULAÇÃO DE DISTORÇÕES DE
PEÇAS FUNDIDAS**

Proposta de dissertação submetida ao programa de Pós-Graduação em Engenharia Mecânica, PPGEM, do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como parte dos requisitos necessários para a obtenção do Título de Mestre.

Orientador: Marcus Vinícius Canhoto Alves

BANCA EXAMINADORA

Dr. Marcus Vinícius Canhoto Alves

Universidade do Estado de Santa Catarina – UDESC/CCT

Membros:

Dr. Marcelo Matos Martins

Green Hydrogen Collaborative Laboratory - HYLAB/SINES

Dr. Carlos Vinicius Opelt

Universidade do Estado de Santa Catarina – UDESC/CCT

Joinville, 31/01/2025

RESUMO

A fundição em moldes de areia é amplamente utilizada na fabricação de peças metálicas com geometrias complexas, sendo fundamental assegurar a precisão dimensional para garantir a qualidade dos produtos fabricados. Durante o resfriamento, as peças fundidas experimentam alterações dimensionais consideráveis devido à contração do metal, o que resulta em uma geometria final que se desvia da forma e dimensão projetadas inicialmente. Para prever e ajustar essas distorções, a indústria tradicionalmente emprega métodos de simulação convencionais, como softwares baseados em diferenças finitas ou elementos finitos, que, embora eficazes, implicam em custos significativos relacionados à aquisição e operação do software. A intensa necessidade de capacidade computacional, exige máquinas potentes e longos períodos de processamento.

Diante desse cenário, o presente trabalho propõe uma abordagem alternativa para a simulação das compensações dimensionais necessárias em peças fundidas. A inovação reside no uso de uma rede neural convolucional auto-encoder, projetada para ajustar as geometrias no estado original, fornecendo uma previsão da geometria em seu estado deformado, correspondente ao comportamento do fundido. A rede é treinada com dados de geometrias cujo comportamento dimensional foi calculado no software Magmasoft. Essa abordagem permite que a rede aprenda a correlação entre a geometria nominal e a deformada, possibilitando a previsão das alterações dimensionais de maneira eficiente e econômica.

A rede neural convolucional com arquitetura auto-encoder demonstrou eficácia na previsão das alterações dimensionais das peças fundidas devido à contração do metal, apresentando erros mínimos na altura, largura e espessura da maioria das geometrias. Os fenômenos de distorção causados pela restrição da areia e pelo resfriamento não uniforme também foram reproduzidos, embora com maior índice de erro. A previsão por geometria foi significativamente mais rápida, levando apenas 1 segundo usando a rede neural treinada, em comparação aos 62 minutos em média exigidos pelas simulações convencionais, evidenciando a eficiência computacional do método proposto.

Os resultados confirmam que a técnica de simulação baseada em aprendizado de máquina é uma alternativa viável e eficiente aos métodos tradicionais. Apesar de limitações relacionadas à qualidade dos dados de treinamento e à complexidade de

certos fenômenos, a rede neural conseguiu reproduzir os principais comportamentos do metal durante a solidificação.

Palavras-chave: Inteligência Artificial; Fundição em areia; Redes neurais; Simulação computacional; Compensações dimensionais.

ABSTRACT

Sand mold casting is widely used in the manufacturing of metallic components with complex geometries, making it essential to ensure dimensional accuracy for product quality. During cooling, cast parts undergo significant dimensional changes due to metal shrinkage, resulting in a final geometry that deviates from the originally designed shape and dimensions. To predict and correct these distortions, traditional industrial methods utilize conventional simulation software, such as finite difference or finite element-based approaches, which, although effective, involve significant costs related to software acquisition and operation. Additionally, these methods require robust computational hardware and extensive processing times.

Considering this scenario, the present work proposes an alternative approach for simulating the dimensional compensations necessary in cast parts. The innovation lies in the use of a convolutional neural network with an autoencoder architecture designed to adjust original geometries, providing a prediction of their deformed state corresponding to casting behavior. The neural network is trained with geometric data calculated using Magmasoft software, enabling it to learn the correlation between nominal and deformed geometries efficiently and economically.

The convolutional neural network with autoencoder architecture demonstrated effectiveness in predicting dimensional changes in cast parts due to metal shrinkage, showing minimal errors in height, width, and thickness for most geometries evaluated. Distortion phenomena caused by sand mold constraints and non-uniform cooling were also reproduced, though with higher error rates. Prediction per geometry was significantly faster, taking only 1 second using the trained neural network, compared to an average of 62 minutes required by conventional simulations, highlighting the computational efficiency of the proposed method.

The results confirm that the machine learning-based simulation technique is a viable and efficient alternative to traditional methods. Despite limitations related to the quality of training data and the complexity of certain phenomena, the neural network successfully replicated the main behaviors of the metal during solidification.

Keywords: Artificial Intelligence; Computer Aided Simulation; Sand Casting; Convolutional Neural Networks Auto-encoder; Pattern Allowances.

LISTA DE ILUSTRAÇÕES

Figura 1 – Comportamentos causadores de distorção em peças fundidas mais estudados	18
Figura 2 – Imagem mostrando a diferença dimensional entre o molde preenchido e a peça final resfriada.	20
Figura 3 – Grãos de areia no estado inicial (a) e após o aquecimento e dilatação (b)	21
Figura 4 - Efeito do resfriamento não uniforme na deformação de uma geometria hipotética	22
Figura 5 – Ilustração da operação de convolução.....	37
Figura 6 – Exemplo de cálculo do erro BCE.....	44
Figura 7 – Exemplo de aplicação do <i>Max Pooling</i>	47
Figura 8 – Exemplo de operação de convolução transposta.....	49
Figura 9 – Arquitetura de uma U-net	50
Figura 10 - Aumento dimensional após a voxelização	56
Figura 11 – ilustração dos verdadeiros e falsos positivos e negativos	59
Figura 12 – Avaliação da previsão do ponto de vista da exatidão.....	61
Figura 13 - Avaliação da previsão do ponto de vista da precisão	62
Figura 14 - Avaliação da previsão do ponto de vista da sensibilidade	63
Figura 15 – Avaliação da previsão do ponto de vista do <i>F1-score</i>	64
Figura 16 – Fluxograma do projeto	66
Figura 17 – Geometrias usadas para treinamento da rede neural	67
Figura 18 – Configuração para simulação de uma das geometrias	69
Figura 19 – Curva da condutância térmica de contato em função da temperatura ...	70
Figura 20 – Aumento de dados por rotação em torno do eixo Z	71
Figura 21 – Fluxograma do programa de multiplicação das geometrias para aumento de dados.....	72
Figura 22 – Fluxograma do programa de pré-processamento dos dados.....	73
Figura 23 – Voxelização da geometria número 3	75
Figura 24 – Arquitetura da rede neural.....	77
Figura 25 – Fluxograma da função de carregamento dos dados	79
Figura 26 – Fluxograma da função de exportação da rede neural	80
Figura 27 - Comparação da redução de comprimento das geometrias 1, 2, 3 e 4....	86

Figura 28 - Geometria nº 5, 15, 16 e 17 nos estados deformados calculados pelo Magmasoft	87
Figura 29 - Estados deformados das geometrias 6, 7 e 8	87
Figura 30 - Geometrias 18 e 19 nos estados deformados	88
Figura 31 - Análise da voxelização da geometria número 4.....	89
Figura 32 - Curvas de erro de treinamento, validação e teste versus época de treinamento.....	90
Figura 33 - Curva de F1-score ao longo das épocas	92
Figura 34 – F1-score por geometria na época 26	93
Figura 35 – Gráfico do F1-score para cada geometria em diferentes graus de refinamento.....	95
Figura 36 - Geometria número 13 calculada pela rede neural da época 26.....	96
Figura 37 – Histograma dos desvios dimensionais da geometria 13	96
Figura 38 – Geometria número 15 calculada pela rede neural da época 26.....	97
Figura 39 – Histograma dos desvios dimensionais da geometria 15	98
Figura 40 – Geometria número 8 calculada pela rede neural da época 26.....	99
Figura 41 – Histograma dos desvios dimensionais da geometria 8	100
Figura 42 - Geometria número 132 calculada pela rede neural da época 26.....	101
Figura 43 - Histograma dos desvios dimensionais da geometria 132	102
Figura 44 - Geometria número 5 calculada pela rede neural da época 26.....	103
Figura 45 - Histograma dos desvios dimensionais da geometria 5	104
Figura 46 - Geometria número 1 calculada pela rede neural da época 26.....	105
Figura 47 - Histograma dos desvios dimensionais da geometria 1	105

LISTA DE TABELAS

Tabela 1 – Composição química do material utilizado	68
Tabela 2 – Geometrias separadas para teste	74
Tabela 3 – Medição dos recursos computacionais utilizados na simulação.....	106

SUMÁRIO

1	INTRODUÇÃO	15
1.1	JUSTIFICATIVA	16
1.2	OBJETIVOS	16
1.2.1	Objetivos gerais.....	16
1.2.2	Objetivos específicos.....	17
2	REVISÃO BIBLIOGRÁFICA	18
2.1	ALTERAÇÕES DIMENSIONAIS EM PEÇAS FUNDIDAS.....	18
2.1.1	Contração térmica e restrição da areia	19
2.1.2	Dilatação e descompactação das paredes do molde	20
2.1.3	Deformações causadas pelo resfriamento não uniforme	22
2.2	MODELAGEM MATEMÁTICA DA FUNDIÇÃO EM AREIA	23
2.2.1	Modelagem das etapas do processo de fundição	23
2.2.1.1	<i>Equação da dinâmica dos fluidos</i>	<i>24</i>
2.2.1.2	<i>Equação da continuidade</i>	<i>25</i>
2.2.1.3	<i>Equação da conservação da energia</i>	<i>25</i>
2.2.1.4	<i>Modelagem da Solidificação e Previsão de Defeitos.....</i>	<i>26</i>
2.2.2	Modelagem matemática de tensões e deformações na simulação computacional da fundição	27
2.2.2.1	<i>Integração com a Simulação Térmica e de Solidificação</i>	<i>28</i>
2.2.2.2	<i>Equações Governantes do Problema Mecânico</i>	<i>28</i>
2.2.2.3	<i>Relações Constitutivas do Metal</i>	<i>29</i>
2.2.2.3.1	Elasticidade Linear (Lei de Hooke Generalizada)	29
2.2.2.3.2	Plasticidade e Fluência.....	30
2.2.2.3.3	Expansão Térmica e Mudanças de Fase	32
2.2.2.4	<i>Não Linearidades na Modelagem.....</i>	<i>32</i>
2.2.2.4.1	Não Linearidade de Material	32
2.2.2.4.2	Não Linearidade de Contato.....	33
2.2.2.4.3	Não Linearidade Temporal	34
2.2.2.5	<i>Métodos Numéricos Aplicados</i>	<i>35</i>
2.3	REDES NEURAS	36
2.3.1	Redes Neurais Convolucionais	36
2.3.2	Funções de ativação	38

2.3.2.1	<i>ReLU (Rectified Linear Unit)</i>	38
2.3.2.2	<i>Leaky ReLU</i>	39
2.3.2.3	<i>Sigmoide</i>	40
2.3.3	Treinamento da rede neural	40
2.3.3.1	<i>Algoritmos de otimização</i>	41
2.3.3.1.1	Otimizador AdamW	41
2.3.3.2	<i>Função de erro BCE</i>	43
2.3.4	Técnicas de regularização	44
2.3.4.1	<i>Dropout</i>	45
2.3.5	Normalização em lote	45
2.3.6	Técnicas de redução e ampliação dimensional	46
2.3.6.1	<i>Max pooling</i>	47
2.3.6.2	<i>Convolução transposta</i>	48
2.3.7	Arquiteturas U-net	50
2.3.8	Redes convolucionais para dados tridimensionais	51
2.4	PROCESSAMENTO DE IMAGENS TRIDIMENSIONAIS	55
2.4.1	Definição de Voxel	55
2.4.2	Voxelização	55
2.4.3	Métricas de Desempenho na Classificação Binária de Voxels com Inteligência Artificial	57
2.4.3.1	<i>Definição de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos</i>	58
2.4.3.2	<i>Exatidão</i>	60
2.4.3.3	<i>Precisão</i>	61
2.4.3.4	<i>Sensibilidade</i>	62
2.4.3.5	<i>F1-Score</i>	64
3	METODOLOGIA	66
3.1	SELEÇÃO E PREPARAÇÃO DAS GEOMETRIAS	66
3.1.1	Projeto das geometrias iniciais	67
3.1.2	Obtenção das geometrias deformadas	68
3.1.3	Programa para aumento dos dados de treinamento	71
3.1.4	Programa de pré-processamento dos dados	72
3.1.5	Divisão dos dados	76
3.2	PROGRAMA DE TREINAMENTO DA REDE NEURAL	76

3.2.1	Arquitetura da rede neural	76
3.2.2	Otimização e Função de Erro	78
3.2.3	Carregamento dos dados sob demanda	79
3.2.4	Função para exportar rede neural	80
3.2.5	Recursos computacionais	81
3.3	PÓS-PROCESSAMENTO E MEDIÇÃO DA PRECISÃO DIMENSIONAL	81
3.3.1	Avaliação do conjunto de testes	81
3.3.1.1	<i>Seleção da Melhor Época de Treinamento</i>	82
3.3.2	Avaliação Dimensional com CloudCompare	83
3.3.2.1	<i>Análise Estatística por Histograma de Desvios</i>	84
3.3.2.2	<i>Análise de sensibilidade ao refinamento da voxelização</i>	84
4	RESULTADOS	85
4.1	ANÁLISE DOS RESULTADOS DA SIMULAÇÃO DO MAGMASOFT	85
4.1.1	Análise da reprodutibilidade do efeito da restrição da areia na contração das geometrias	85
4.1.2	Análise da reprodutibilidade do efeito da restrição da areia na distorção das geometrias	86
4.1.3	Análise da reprodutibilidade do efeito do resfriamento não uniforme	87
4.1.4	Análise da reprodutibilidade do efeito da dilatação do molde	88
4.2	ANÁLISE DO PRÉ-PROCESSAMENTO DAS GEOMETRIAS	88
4.3	RESULTADOS RELACIONADOS AO TREINAMENTO DA REDE NEURAL	89
4.3.1	Análise do comportamento do erro	90
4.3.2	Análise do comportamento do F1-score	91
4.4	AVALIAÇÃO DA PERFORMANCE DA REDE NO CONJUNTO DE TESTES	92
4.4.1	Avaliação para seleção da melhor época	93
4.4.2	<i>Impacto do Tamanho dos Voxels sobre o F1-Score</i>	94
4.4.3	Análise das geometrias tesselladas	95
4.4.3.1	<i>Geometria número 13</i>	95
4.4.3.2	<i>Geometria número 15</i>	97
4.4.3.3	<i>Geometria número 8</i>	99
4.4.3.4	<i>Geometria número 132</i>	100

4.4.3.5	<i>Geometria número 5</i>	102
4.4.3.6	<i>Geometria número 1</i>	104
4.4.4	Utilização de recursos computacionais – Simulação convencional	106
4.4.5	Utilização de recursos computacionais – Redes neurais	107
5	CONCLUSÕES	108
5.1	DESEMPENHO GERAL DA REDE NEURAL.....	108
5.2	LIMITAÇÕES IDENTIFICADAS.....	108
5.3	INFLUÊNCIA DO GRAU DE REFINAMENTO	109
5.4	COMPARAÇÃO COM A SIMULAÇÃO CONVENCIONAL	109
5.5	REQUISITOS PARA APLICAÇÃO INDUSTRIAL DO SOFTWARE	110
	REFERÊNCIAS	111
	APÊNDICE A – PROPRIEDADES MECÂNICAS E FÍSICAS DO METAL UTILIZADO	117
	APÊNDICE B – PROPRIEDADES DA AREIA UTILIZADA	122
	APÊNDICE C – FLUXOGRAMA DO PROGRAMA DE TREINAMENTO	125
	APÊNDICE D – ARQUITETURA DA REDE NEURAL	128
	APÊNDICE E – FLUXOGRAMA DO PROGRAMA DE TESTES	129

1 INTRODUÇÃO

A fundição em moldes de areia permanece como uma das principais tecnologias empregadas na fabricação de componentes metálicos de geometrias complexas, sendo amplamente utilizada nos setores automotivo, aeroespacial e de bens de capital. Entretanto, apesar de sua versatilidade e viabilidade econômica, alcançar precisão dimensional nas peças fundidas continua sendo um desafio técnico significativo, especialmente devido às deformações térmicas e mecânicas que ocorrem durante o resfriamento e solidificação do metal (Campbell, 2011).

Tradicionalmente, esses desafios têm sido enfrentados por meio de simulações computacionais baseadas em modelos termomecânicos, frequentemente implementados através do método dos elementos finitos. Trabalhos importantes, como o de Galles e Beckermann (2017), demonstraram que distorções em peças fundidas são fortemente influenciadas tanto pelas restrições impostas pelo molde durante o processo de resfriamento quanto pela dilatação da areia do molde nos instantes iniciais do seu preenchimento, sendo imprescindível considerar os efeitos de dilatação do molde e restrições mecânicas para prever corretamente as alterações dimensionais. Da mesma forma, Baghani et al. (2014) apontaram que o uso de modelos termomecânicos acoplados permite prever tensões residuais em função das restrições do molde e da taxa de resfriamento.

Recentemente, os avanços em inteligência artificial (IA) têm oferecido novas possibilidades para aprimorar esses métodos convencionais. Diversas aplicações de aprendizado de máquina têm demonstrado resultados positivos na previsão de defeitos, otimização de processos e controle de qualidade em fundição. Bhagwat et al. (2024) revisaram o uso de técnicas como redes neurais, máquinas de vetores de suporte e algoritmos genéticos para otimizar processos produtivos, melhorar o controle de qualidade e reduzir defeitos. Breznikar et al. (2024), por exemplo, aplicaram o algoritmo k-vizinhos mais próximos (k-NN) para reduzir defeitos relacionados à mistura de areia bentonítica, resultando em classificadores capazes de prever, com alta precisão, parâmetros tecnológicos críticos para a qualidade do produto final.

Apesar desses avanços significativos, permanece uma lacuna relevante na aplicação direta de métodos de IA para prever alterações dimensionais específicas em processos de fundição em areia. Este trabalho busca preencher essa lacuna, desenvolvendo um modelo preditivo baseado em redes neurais artificiais, treinado com dados reais geométricos e termomecânicos do processo de fundição em areia.

Este trabalho está estruturado da seguinte forma: após esta introdução, o Capítulo 2 aborda a revisão bibliográfica, detalhando as bases teóricas de alterações dimensionais em fundição e de redes neurais. O Capítulo 3 descreve a metodologia empregada no ensaio, explicando como foram elaboradas as geometrias de treinamento e testes, e como foi desenvolvido e executado o programa responsável pelas simulações. O Capítulo 4 apresenta os resultados obtidos em todas as etapas do teste realizado, seguido pelo Capítulo 5, que traz as considerações finais, discutindo os resultados e sugerindo possíveis aplicações futuras da técnica desenvolvida.

1.1 JUSTIFICATIVA

A indústria de fundição lida frequentemente com previsão de compensações dimensionais em ferramentais, afetando tempo e custos no desenvolvimento de novos produtos. Este trabalho introduz a inteligência artificial como uma nova abordagem para a previsão de variações dimensionais em ferramentais de fundição, temas que, até o momento, têm recebido atenção limitada na literatura. A aplicação de técnicas de Inteligência Artificial – IA, oferece uma perspectiva inovadora para enfrentar esta questão, prometendo avanços significativos na precisão e na redução de custos no processo de desenvolvimento de produtos na indústria de fundição.

1.2 OBJETIVOS

1.2.1 Objetivos gerais

Desenvolver, treinar e validar uma rede neural convolucional tridimensional capaz de prever as deformações geométricas de peças

submetidas ao processo de fundição em moldes de areia e avaliar sua viabilidade como alternativa aos métodos de simulação convencionais.

1.2.2 Objetivos específicos

- Projetar e implementar uma arquitetura de rede neural convolucional 3D adequada à previsão de distorções em geometrias fundidas, utilizando como entrada e saída geometrias voxelizadas no formato STL.
- Gerar e preparar uma base de dados de geometrias deformadas, representando diferentes comportamentos físicos relevantes da fundição em areia, como contração térmica, restrição do molde e Resfriamento não uniforme.
- Avaliar a precisão do modelo treinado por meio de métricas quantitativas, como erro binário (BCE) e F1-score, e analisar sua robustez em relação à variação do grau de refinamento espacial.
- Comparar o desempenho da rede neural com o de simulações convencionais em termos de acurácia, escalabilidade e tempo de processamento.
- Identificar as limitações da abordagem proposta e discutir os requisitos para sua aplicação em escala industrial.

2 REVISÃO BIBLIOGRÁFICA

2.1 ALTERAÇÕES DIMENSIONAIS EM PEÇAS FUNDIDAS

Na fabricação de peças metálicas por meio da fundição em areia, é comum observar divergências dimensionais nas peças resfriadas em relação à geometria original da cavidade do molde. Porém, prever precisamente o comportamento dessas alterações ainda permanece um desafio. Diversos fenômenos contribuem para as distorções observadas na geometria final das peças, sendo que entre os mais discutidos na literatura destacam-se quatro: a contração térmica do metal, a resistência oferecida pela areia à contração da peça, a expansão das paredes do molde quando em contato com o metal em alta temperatura e o resfriamento não uniforme. (Galles; Hardin; Beckermann, 2012)

Figura 1 – Comportamentos causadores de distorção em peças fundidas mais estudados



Fonte: Elaborada pelo autor (2025)

Conforme ilustra a Figura 1, a alteração dimensional de peças fundidas é causada por diferentes comportamentos que ocorrem durante o resfriamento da peça, cuja combinação resulta na alteração dimensional total de uma peça. Além desses quatro comportamentos, é possível que ocorram ainda outros não abordados neste trabalho, a depender da liga do metal, da composição da areia e ligantes, da forma de montagem de núcleos de areia no molde e etc. (Galles; Beckermann, 2017a)

Nos subcapítulos subsequentes, será feita uma análise das bases teóricas de alguns desses fenômenos, buscando explicar seus efeitos no resultado da fundição.

2.1.1 Contração térmica e restrição da areia

A contração térmica e a restrição da areia são conceitos fundamentais na fundição de metais que determinam a precisão dimensional e a integridade estrutural das peças fundidas. A dilatação térmica, ou "contração térmica" como referido neste contexto, é o fenômeno pelo qual as dimensões de um corpo mudam em resposta a variações na temperatura. (Saada; Bonnet; Bouvard, 1996)

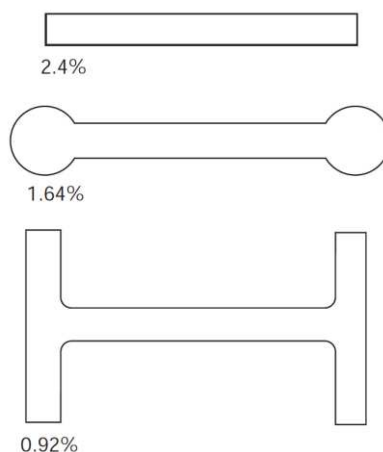
Assim que o metal líquido é vazado em um molde de areia, este começa a resfriar e, conseqüentemente, a contrair-se. Inicialmente, a contração do metal não altera imediatamente as dimensões finais da peça fundida, já que o metal estando ainda líquido possibilita a movimentação e redistribuição do fluido. É apenas após a solidificação que a contração começa a influenciar na geometria final. (Campbell, 2011)

A contração do metal segue um padrão relativamente previsível, contanto que o metal esteja livre de restrições. No entanto, a presença do molde de areia introduz uma variável importante: a restrição. Esta restrição não é constante, mas varia significativamente em função da geometria da peça, da composição e da compactação da areia, assim como da interação entre diferentes seções da peça e o molde. Essas interações resultam em uma contração não uniforme, que pode causar deformações ou tensões internas na peça fundida, afetando sua integridade estrutural e precisão dimensional. (Baghani *et al.*, 2014)

A Figura 2 mostra três geometrias com diferentes graus de restrição da areia. Segundo Campbell (2011), o grau de restrição proporcionado pelo molde de areia afeta a contração global do metal durante o resfriamento, dessa forma, a barra mostrada na Figura 2, que não sofre restrição da areia, contrai livremente alcançando redução no comprimento mais que duas vezes a redução observada na terceira geometria mostrada na Figura 2. Em condições de alta restrição, o molde impede a contração livre, forçando o metal a se esticar levemente,

enquanto em situações de restrição parcial a peça pode apresentar deformações plásticas, com regiões submetidas a tensão ou compressão excessiva.

Figura 2 – Imagem mostrando a diferença dimensional entre o molde preenchido e a peça final resfriada.



Fonte: (Campbell, 2011)

Esta restrição é particularmente relevante durante os momentos iniciais do resfriamento, quando o metal ainda está relativamente quente e sua resistência mecânica é baixa. A complexidade da interação metal-molde é acentuada pela variação das propriedades mecânicas da areia com a temperatura. (Galles; Beckermann, 2016)

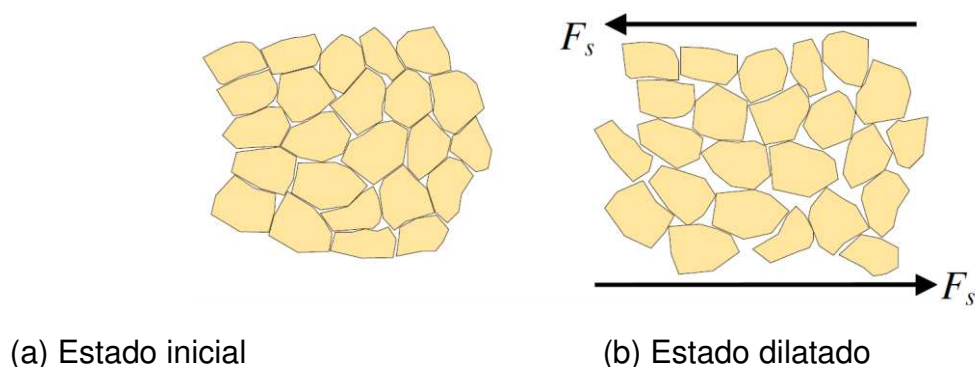
2.1.2 Dilatação e descompactação das paredes do molde

A diferença de temperatura entre a areia e o metal induz uma forte transferência de calor entre eles, a qual eleva a temperatura da areia que compõe a cavidade do molde. Este aumento de temperatura da areia promove sua dilatação, que, por sua vez, causa o deslocamento das paredes do molde já nos primeiros segundos após o início do preenchimento. Devido ao metal estar, em sua maior parte, líquido quando ocorre o deslocamento das paredes do molde, todo o deslocamento das paredes do molde é repassado ao metal e carregado para a peça final, apresentando diferenças dimensionais. (Svidró; Diószegi; Svidró, 2020)

Thiel e Ravi (2016) investigaram como o aquecimento da areia do molde, causado pela transferência de calor do metal fundido, impacta as dimensões

finais da peça fundida. Eles observaram que, em regiões mais espessas do metal, onde o tempo de solidificação é maior, a temperatura da areia adjacente aumenta significativamente, frequentemente ultrapassando 1.300°C. Este aquecimento eleva a expansão térmica da areia, especialmente durante a transição do quartzo alfa para quartzo beta e a possível formação de cristobalita em temperaturas mais altas. A expansão do núcleo de areia desloca as paredes do molde enquanto o metal ainda está líquido ou em transição de fase, resultando em deformações que são transferidas para a peça fundida. Além disso, regiões com maior massa de metal tendem a amplificar esse efeito, devido à maior retenção de calor e ao aumento da mobilidade do metal, contribuindo para diferenças dimensionais mais pronunciadas nas áreas adjacentes do fundido. (Thiel; Ravi, 2016)

Figura 3 – Grãos de areia no estado inicial (a) e após o aquecimento e dilatação (b)



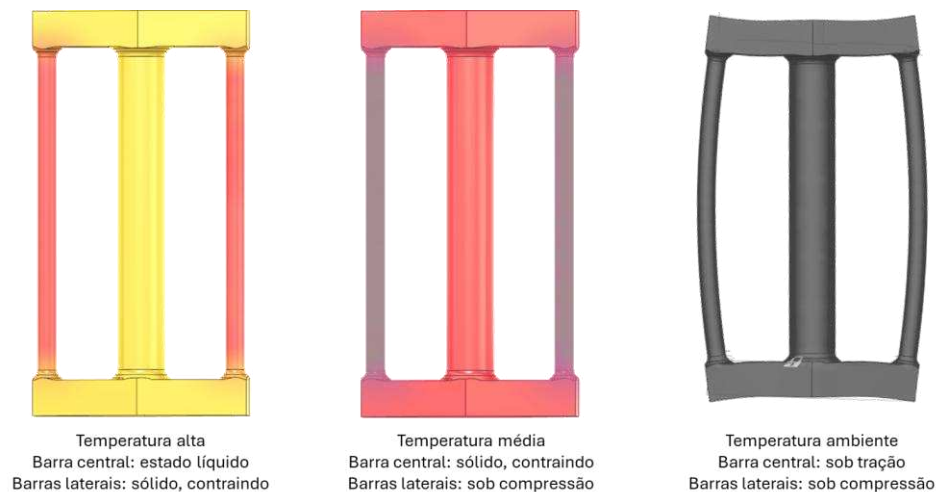
Fonte: (Galles; Beckermann, 2017a)

Galles; Beckermann (2017a) em seu estudo sobre distorções e ajustes de padrões durante a fundição em areia de aço, descreveram o fenômeno de dilatação da areia, ilustrado na Figura 3. Esse efeito ocorre quando a areia do molde é exposta ao calor intenso do metal fundido, resultando em uma expansão térmica que aumenta o volume local da matriz granular. Durante este processo, o rearranjo dos grãos provoca uma dilatação mecânica adicional, que pode levar ao deslocamento das paredes do molde.

2.1.3 Deformações causadas pelo resfriamento não uniforme

Campbell (2011) destaca o impacto das deformações causadas pelo resfriamento não uniforme no comportamento dimensional das peças fundidas. Esse fenômeno ocorre quando diferentes regiões de uma peça solidificam em momentos distintos, resultando em tensões internas devido às variações de contração térmica entre as áreas já solidificadas e as que ainda permanecem no estado líquido ou em transição de fase.

Figura 4 - Efeito do resfriamento não uniforme na deformação de uma geometria hipotética



Fonte: Elaborada pelo autor (2025).

A Figura 4 mostra uma geometria hipotética composta por regiões mais finas e outras mais espessas. As regiões mais finas e/ou mais expostas à transferência de calor tendem a solidificar antes e já iniciam a contração e redução dimensional. Nesse momento, essas regiões têm pouca restrição a deformação, já que o restante da peça está em temperaturas mais altas e, portanto, com menor resistência mecânica. Num momento posterior as regiões mais espessas e/ou com uma taxa de transferência de calor menor adquirem maior coesão e iniciam a contração e redução dimensional. A contração da região central da geometria mostrada na Figura 4 submete as barras laterais a esforços de compressão, podendo ocasionar a flambagem delas. (Campbell, 2011)

Essas distorções são particularmente pronunciadas em peças de geometria complexa ou com variações significativas de espessura, onde o fluxo de calor e o tempo de solidificação são heterogêneos. Campbell enfatiza a importância de um projeto cuidadoso de moldes e o uso de técnicas como a aplicação de materiais refratários ou a modificação da geometria do molde para minimizar essas deformações e melhorar a precisão dimensional. (Campbell, 2011)

2.2 MODELAGEM MATEMÁTICA DA FUNDIÇÃO EM AREIA

A modelagem matemática desempenha um papel fundamental na análise e otimização do processo de fundição em areia, fornecendo a base para os programas computacionais que simulam o comportamento do metal desde o preenchimento da cavidade até a solidificação final da peça. Nesta seção são abordadas inicialmente as etapas principais do processo, incluindo a dinâmica dos fluidos, continuidade, conservação da energia e a modelagem da solidificação, essenciais para prever defeitos estruturais nas peças fundidas. (Şensoy, 2022)

Posteriormente, explora-se a simulação das tensões e deformações integradas às análises térmicas, destacando as equações governantes e as relações constitutivas do material, como elasticidade, plasticidade, fluência e expansões térmicas. Além disso, são discutidas as complexidades relacionadas às não linearidades de material, contato e temporais, bem como os métodos numéricos aplicados para resolver esses desafios computacionais na simulação da fundição.

2.2.1 Modelagem das etapas do processo de fundição

A simulação computacional dos processos de fundição é uma ferramenta essencial para prever e otimizar o comportamento do metal líquido durante o preenchimento do molde e sua subsequente solidificação. De acordo com Hattel (2005), a modelagem de fundição pode ser dividida em três etapas principais:

1. Preenchimento do molde: Nesta fase, o metal líquido é introduzido na cavidade do molde, e seu comportamento é descrito pelas equações de escoamento de fluidos.
2. Solidificação: Após o preenchimento, o metal começa a solidificar. Este processo é governado pela equação de energia e por modelos de solidificação que descrevem a formação da microestrutura do material.
3. Análise de tensões: Finalmente, é realizada uma análise para prever as tensões residuais e possíveis distorções na peça fundida, garantindo a integridade estrutural do produto.

A solução dessas equações requer a aplicação de métodos numéricos como Método de Diferenças Finitas (FDM), o Método dos Volumes Finitos (FVM) e o Método dos Elementos Finitos (FEM)

2.2.1.1 Equação da dinâmica dos fluidos

Durante o preenchimento do molde, o metal líquido flui e resfria ao mesmo tempo, em um processo governado pelas equações de dinâmica dos fluidos que relacionam massa, quantidade de movimento e energia. A forma diferencial da equação de Navier–Stokes (Equação 1) é fundamental para descrever a conservação da quantidade de movimento do fluido, levando em conta a inércia, a viscosidade, os gradientes de pressão e as forças externas, como apresentado na equação (1). (Hattel, 2005)

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{f} \quad (1)$$

Onde:

- ρ : é a densidade
- \mathbf{v} : o campo de velocidade
- p : a pressão
- μ : a viscosidade dinâmica
- \mathbf{f} : as forças volumétricas (por exemplo, a gravidade).

Nos processos de fundição, o metal líquido costuma ter elevada densidade e alta temperatura, resultando em condições de escoamento que podem variar de laminar a transicional ou até fortemente turbulento, dependendo do projeto do sistema de canais de alimentação, da taxa de vazamento e das propriedades termofísicas do metal. Em regimes com número de Reynolds elevado, os termos convectivos passam a exercer influência dominante, enquanto a viscosidade dinâmica e a condução de calor têm maior relevância nos estágios iniciais ou em zonas de escoamento lento. (Versteeg & Malalasekera, 2007).

2.2.1.2 Equação da continuidade

Além disso, a equação de Navier–Stokes deve ser acoplada à equação de continuidade, mostrada na equação (2), para assegurar a conservação de massa. Essa equação impõe que quaisquer variações locais na densidade, que podem ocorrer devido à variação de temperatura, sejam compensadas pelo fluxo de massa, assegurando que o campo de velocidade obtido seja fisicamente consistente. Em muitas aplicações de fundição, considera-se o escoamento quase incompressível, pois a variação de densidade com pressão é pequena em comparação com a variação de densidade devida à temperatura. A temperatura afeta diretamente a viscosidade e a densidade do metal, tornando a transferência de calor um ponto importante para descrever corretamente o comportamento do fluido (Bird, Stewart & Lightfoot, 2002).

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2)$$

Onde:

- ρ : é a densidade
- \mathbf{v} : o campo de velocidade
- t : tempo

2.2.1.3 Equação da conservação da energia

Paralelamente, a equação de energia, mostrada na equação (3), é utilizada para descrever a transferência de calor durante o processo de fundição. Através dessa equação é determinada a distribuição de temperatura na peça à medida

que ela resfria, influenciando propriedades termofísicas como a viscosidade e a densidade. Dessa forma, a equação de energia cria um ciclo com as equações de Navier–Stokes e continuidade, já que as variações de temperatura alteram as propriedades do fluido, afetando o escoamento (Chen et al., 2022).

$$\rho c_p \frac{\partial T}{\partial t} = k \nabla^2 T + \dot{q} \quad (3)$$

Onde:

- ρ : é a densidade
- c_p : calor específico
- T : temperatura
- t : tempo
- k : condutividade térmica
- \dot{q} : taxa de liberação de calor latente durante a solidificação

2.2.1.4 Modelagem da Solidificação e Previsão de Defeitos

Após o preenchimento do molde, o processo de solidificação passa a exercer influência dominante na formação da peça fundida. Nesse estágio, a equação de conservação de energia torna-se primordial, pois o resfriamento não uniforme do metal induz gradientes térmicos que aumentam a complexidade da distribuição de calor na peça. Para uma previsão correta, é imprescindível incorporar modelos de solidificação à equação de energia, destacando-se a Regra da Alavanca e o Modelo de Scheil. Essas abordagens permitem estimar a fração sólida (e conseqüentemente o calor latente liberado) em função da temperatura e da composição local do metal. (Hattel, 2005)

A Regra da Alavanca baseia-se em um regime de equilíbrio local entre as fases sólida e líquida, supondo difusão completa tanto no líquido quanto no sólido. Com isso, a fração sólida a cada instante de temperatura pode ser determinada pela proporção de elementos de liga que se distribuem entre as fases, de acordo com o diagrama de equilíbrio. Esse modelo é particularmente adequado quando as velocidades de solidificação são mais lentas e há tempo suficiente para que ocorra difusão significativa no sólido. (Bertelli, 2012)

Por outro lado, o Modelo de Scheil adota uma condição de não equilíbrio mais rigorosa, assumindo ausência de difusão no sólido e difusão instantânea no líquido. Como resultado, a fração sólida e a distribuição de solutos na fase sólida são calculadas progressivamente à medida que a solidificação avança, levando a uma segregação mais acentuada dos elementos de liga. O modelo de Scheil é especialmente útil para prever a microestrutura em processos de resfriamento mais rápidos, onde a difusão no sólido é limitada. (Bertelli, 2012)

A modelagem precisa do termo \dot{q} – associado à liberação de calor latente – é crucial para determinar a posição e a extensão da região de solidificação parcial (*mushy zone*), onde coexistem fases líquida e sólida. A compreensão detalhada dessa região é fundamental para prever a formação de microestruturas e a possibilidade de defeitos como porosidades e rechupes. (Chen *et al.*, 2022)

Diversos estudos experimentais e numéricos têm sido realizados para validar a aplicação dessas teorias de solidificação e suas capacidades preditivas de defeitos. Como exemplo, Verran *et al.* (2002) conduziram análises comparativas entre simulações numéricas e resultados experimentais, avaliando a influência da velocidade de enchimento na formação de imperfeições em ligas de alumínio fundidas sob pressão. Os resultados corroboraram o potencial das simulações numéricas para prever defeitos como solda fria e porosidades, permitindo a otimização dos parâmetros de processo e a consequente redução de falhas na peça final. (Verran; Rebello; Oliveira, 2000)

2.2.2 Modelagem matemática de tensões e deformações na simulação computacional da fundição

A previsão de distorções e tensões residuais em peças fundidas é um desafio bastante complexo da modelagem numérica de processos de fundição. Tais tensões decorrem dos fenômenos físicos abordados nas seções anteriores (resfriamento não uniforme, solidificação, mudanças de fase, etc.) e precisam ser representadas de forma consistente para garantir que as simulações reflitam a realidade do processo. (Edacherian; Abhilash; Joseph, 2009)

Para tanto, segundo Hattel (2005), a simulação mecânica deve considerar:

- Acoplamento térmico-mecânico com a análise de preenchimento e solidificação;
- Não linearidades estruturais, incluindo plasticidade, fluência, contato e dependência das propriedades com o tempo e a temperatura;
- Relações constitutivas adequadas ao comportamento do metal (elasticidade, plasticidade, fluência e mudanças de fase);
- Métodos numéricos específicos para problemas transientes, não lineares e de grande escala, como o Método dos Elementos Finitos (FEM).

Nas próximas seções são discutidos em detalhe os principais aspectos da modelagem matemática usados para prever a deformação e as tensões após a solidificação.

2.2.2.1 Integração com a Simulação Térmica e de Solidificação

A análise de tensões e deformações surge após o término das etapas de preenchimento e solidificação, que fornecem as distribuições de temperatura e de fração de fase (sólido/líquido). Esses campos termofísicos são usados como condições iniciais da análise mecânica, resultando em um problema termo-mecânico transiente, em que a evolução da temperatura afeta diretamente o desenvolvimento de tensões. (Galles; Beckermann, 2017b)

2.2.2.2 Equações Governantes do Problema Mecânico

A análise de tensões em peças fundidas apoia-se na equação de equilíbrio da mecânica dos sólidos, conforme discutido por Zienkiewicz, Taylor e Zhu (2005), cujo formato diferencial genérico é apresentado na Equação (4). Em sua forma simplificada, ela expressa que a soma das forças internas (representadas pelas tensões) e das forças volumétricas deve ser nula em qualquer ponto do corpo.

$$\nabla \cdot \sigma + fb = 0 \quad (4)$$

onde:

- σ é o tensor de tensões,
- f_b representa forças de volume (por exemplo, gravidade).

No contexto de simulação de fundição em areia, essa equação de equilíbrio é aplicada de forma transiente, pois as propriedades do metal (como módulo de elasticidade, condutividade térmica e densidade) variam significativamente com a temperatura e com o estado de solidificação. Além disso, as paredes do molde têm um papel na análise de contato ou restrição da peça em determinadas regiões, influenciando as condições de contorno mecânicas. (Galles; Beckermann, 2017b)

2.2.2.3 Relações Constitutivas do Metal

A análise de tensões e deformações em peças fundidas depende das relações constitutivas do material, que descrevem como o metal responde à aplicação de cargas e às variações de temperatura durante o processo de resfriamento. Em geral, o comportamento mecânico pode ser dividido em três regimes principais: elasticidade, plasticidade/fluência e expansão térmica (associada ou não a mudanças de fase). Cada regime é detalhado nas próximas subseções. (Edacherian; Abhilash; Joseph, 2009; Hattel, 2005)

2.2.2.3.1 Elasticidade Linear (Lei de Hooke Generalizada)

Para pequenas deformações e temperaturas moderadas (normalmente inferiores à temperatura de recristalização), o comportamento elástico do metal pode ser descrito pela elasticidade linear. Nessa abordagem, como mostra a equação (5), as tensões (σ) são proporcionais às deformações (ε) via um tensor de rigidez de quarta ordem (C). (Zienkiewicz; Taylor; Zhu, 2005)

$$\sigma = C : \varepsilon \quad (5)$$

onde:

- σ é o tensor de tensões,
- C é o tensor de rigidez de quarta ordem
- ε é o tensor de deformações

Em muitos metais fundidos, pode-se assumir isotropia (propriedades iguais em todas as direções). Nesse caso, C reduz-se aos módulos de Lamé (λ e μ):

$$\sigma_{ij} = \lambda \delta_{ij} \varepsilon_{kk} + 2 \mu \varepsilon_{ij} \quad (6)$$

onde:

- σ é o tensor de tensões,
- λ e μ são os módulos de Lamé
- δ_{ij} é o delta de Kronecker

Esse regime linear e elástico permanece válido enquanto as tensões não ultrapassarem o limite de escoamento σ_Y , valor que pode mudar significativamente com a temperatura. Em peças fundidas, costuma-se trabalhar com essa lei em etapas de resfriamento iniciais ou em regiões onde o gradiente térmico não é tão acentuado. No entanto, conforme a peça se contrai e as tensões aumentam, podem ocorrer deformações plásticas, exigindo um modelo mais complexo. (Hattel, 2005)

Mesmo dentro do regime elástico, o módulo de elasticidade e o limite de escoamento sofrem redução à medida que a temperatura se eleva. Em fundições de ligas de alumínio, por exemplo, há uma queda notável no módulo de Young se a temperatura estiver próxima do ponto de solidificação. Para se obter maior precisão, é comum utilizar curvas de propriedades dependentes da temperatura (Dowling, 2012).

2.2.2.3.2 Plasticidade e Fluência

Quando as tensões excedem o limite elástico, deve-se levar em conta fenômenos inelásticos como plasticidade e, em regimes de temperatura alta ou tempos longos de solidificação/resfriamento, a fluência. Um modelo comumente usado para representar a plasticidade é baseado na hipótese de endurecimento isotrópico (ou mesmo plasticidade perfeita). A condição de escoamento é dada pela superfície de von Mises, mostrada na equação (7). (Dowling, 2012)

$$\sqrt{\frac{3}{2}} s: s = \sigma_Y \quad (7)$$

onde:

- s é o tensor de tensões desviadoras
- σ_Y é o tensor de tensões,

Em situações de fundição em areia – especialmente quando há grandes variações térmicas – as regiões próximas aos canais de alimentação podem sofrer contrações rápidas e gerar tensões excedentes em zonas específicas, ativando a plasticidade local. Se o metal ultrapassa o limite de escoamento, parte das deformações não será recuperada quando a peça estiver à temperatura ambiente. (Dowling, 2012)

A fluência é uma deformação dependente do tempo, típica de temperaturas elevadas (geralmente acima de $0,4 T_m$, onde T_m é a temperatura de fusão). Ela pode se manifestar durante o resfriamento, se a peça permanece por um período significativo em faixas de temperatura elevadas. A relação constitutiva pode ser modelada por leis que ligam a taxa de deformação ($\dot{\epsilon}_{cr}$) à tensão e à temperatura, como mostrado na equação (8). (Dowling, 2012)

$$\frac{d\epsilon_{cr}}{dt} = f(\sigma, T) \quad (8)$$

onde:

- ϵ_{cr} representa a deformação por fluência
- $f(\sigma, T)$ caracteriza a sensibilidade à temperatura

Nos processos de fundição em que o tempo de resfriamento é relativamente curto, a fluência pode não ser determinante. Porém, em ligas de aço ou em fundição de superligas, onde a peça passa por longos períodos acima de temperaturas críticas, a fluência torna-se relevante (Dowling, 2012).

2.2.2.3.3 Expansão Térmica e Mudanças de Fase

O termo de expansão térmica frequentemente é responsável por boa parte das distorções que se desenvolvem em peças fundidas, pois o metal se contrai ao passar do estado líquido para sólido e ao resfriar até a temperatura ambiente. Essa variação volumétrica é incluída na decomposição de deformação total, mostrada na equação (9). (Dowling, 2012)

$$\varepsilon = \varepsilon_{elástica} + \varepsilon_{plástica} + \varepsilon_{fluência} + \alpha \Delta T \mathbf{I} \quad (9)$$

onde:

- ε_{cr} representa a deformação
- α é o coeficiente de expansão térmica
- \mathbf{I} é o tensor identidade

2.2.2.4 Não Linearidades na Modelagem

A simulação de tensões e deformações em peças fundidas envolve diversos fenômenos físico-químicos que se desenrolam de forma acoplada e transiente. Nesse contexto, é necessário lidar com três principais fontes de não linearidade que incrementam a complexidade – e, por consequência, o custo computacional – dos modelos numéricos. Em aplicações industriais onde a qualidade dimensional e a integridade mecânica da peça são críticas, a representação adequada de cada uma dessas não linearidades é fundamental (Zienkiewicz; Taylor; Zhu, 2005)

2.2.2.4.1 Não Linearidade de Material

Uma das maiores fontes de complexidade na modelagem de fundição está relacionada às propriedades do metal, que variam com a temperatura, grau de solidificação e taxa de resfriamento. As seguintes características são relevantes:

- Plasticidade
O metal pode ultrapassar o limite de escoamento em regiões que sofrem forte contração térmica ou gradiente de temperatura elevado,

desencadeando deformação plástica. Modelos de plasticidade (por exemplo, von Mises com endurecimento isotrópico) requerem iterações adicionais em cada passo de tempo ou de carga, elevando o custo computacional (Dowling, 2012).

- **Fluência**
Em casos de temperaturas elevadas ou longos tempos de solidificação, a deformação dependente do tempo (creep) pode ocorrer. Leis de fluência (Norton, Bailey-Dahms etc.) agregam equações diferenciais extras ao problema, implicando solução não linear acoplada (Bathe, 1996).
- **Dependência Térmica das Propriedades:** O módulo de elasticidade, o limite de escoamento e a viscosidade variam ao longo da solidificação, exigindo modelos termo-dependentes ou tabelas que correlacionem a temperatura às propriedades mecânicas. Cada variação requer reatualização dos parâmetros numéricos a cada incremento de tempo. (Suresh, 2004)

Esses fatores de não linearidade de material demandam algoritmos robustos, como Newton-Raphson não linear ou suas variantes (Modified, Quasi-Newton), para garantir a convergência em cada estágio do problema.

2.2.2.4.2 Não Linearidade de Contato

A interface entre o metal e o molde (por exemplo, moldes de areia ou permanentes) é outro elemento crítico na análise termo-mecânica:

- **Evolução do Contato:** À medida que o metal resfria e se contrai, a separação entre peça e molde pode mudar. Esse efeito altera o coeficiente de transferência de calor, pois a condutividade efetiva na interface tende a reduzir quando há formação de vazios ou fendas. Em contrapartida, em determinados pontos, o metal pode se expandir localmente, aumentando a pressão de contato e modificando as tensões na peça (Campbell, 2011).
- **Condições de Fricção**
Em processos de fundição em areia, costuma-se assumir fricção

desprezível; porém, em moldes metálicos (ou semipermanentes), o atrito na interface pode contribuir para a restrição das deformações térmicas, elevando as tensões residuais. Modelos de contato não linear (penalização, Lagrange, etc.) tornam o problema significativamente mais complexo (Zienkiewicz; Taylor; Zhu, 2005).

- Incerteza nos Parâmetros de Contato
Parâmetros como coeficiente de transferência de calor e coeficiente de atrito podem variar no tempo e depender do histórico térmico, introduzindo incerteza adicional à modelagem. (Kotas, 2011)

Esses efeitos de não linearidade de contato são especialmente importantes em simulações de peças de grande porte ou geometrias complexas, onde o contato metal-molde não é uniforme. (Inoue *et al.*, 2013)

2.2.2.4.3 Não Linearidade Temporal

Toda a evolução do processo de fundição – do preenchimento à solidificação e posterior resfriamento – ocorre de forma transiente, sendo dependente do tempo:

- Variação da Fração Sólida: Durante a solidificação, a fração sólida aumenta gradualmente. A peça passa de um comportamento quase fluido para sólido, e cada estágio possui propriedades bem distintas. Esse processo de transição de fase exige malhas adaptativas ou subrotinas que atualizam as propriedades a cada incremento de tempo. (Bertelli, 2012)
- Alteração do Campo de Temperatura: O campo térmico muda continuamente, afetando a distribuição de tensões e a evolução das deformações. Métodos de integração no tempo (Euler implícito, Crank-Nicolson etc.) são empregados para garantir a estabilidade numérica. (Incropera *et al.*, 2007)
- Acoplamento Térmico-Mecânico: O metal que solidifica e resfria libera ou absorve calor de maneira não linear (calor latente, mudanças de entalpia). Isso cria um feedback entre o problema térmico e o

mecânico, fazendo com que ambos devam ser resolvidos de forma iterativa ou totalmente acoplada. (Galles; Beckermann, 2017b)

A não linearidade temporal eleva o custo computacional na medida em que cada passo de tempo requer a resolução de sistemas de equações termo-mecânicos não lineares, muitas vezes com número considerável de graus de liberdade. (Kotas, 2011)

2.2.2.5 Métodos Numéricos Aplicados

Para resolver o problema termo-mecânico em fundição, o Método dos Elementos Finitos (FEM) é amplamente empregado, oferecendo grande flexibilidade na discretização de geometrias complexas e na consideração de propriedades termo-dependentes. Em sua forma discreta, o sistema de equações que governa o comportamento dinâmico e/ou transiente do corpo pode ser expresso pela Equação (10). (Bathe, 1996; Zienkiewicz; Taylor; Zhu, 2005)

$$\mathbf{M} \frac{d^2 \mathbf{u}}{dt^2} + \mathbf{C} \frac{d\mathbf{u}}{dt} + \mathbf{K}\mathbf{u} = \mathbf{F} \quad (10)$$

onde:

- M é a matriz de massa (mecânica ou termo-mecânica),
- C é a matriz de amortecimento (térmico ou estrutural),
- K é a matriz de rigidez,
- u são os deslocamentos nodais,
- F é o vetor de forças (externas e internas, incluindo as térmicas).

No contexto termo-mecânico, a matriz M pode envolver não apenas termos inerciais (caso de análises dinâmicas), mas também coeficientes que relacionam propriedades térmicas ao comportamento estrutural (por exemplo, modelos acoplados de condução de calor). Já a matriz C pode estar associada a fenômenos dissipativos de natureza mecânica (amortecimento estrutural) ou representar termos de difusão e condução térmica na formulação de elementos finitos para transferência de calor. (Bathe, 1996)

O termo $\mathbf{K}\mathbf{u}$ relaciona as deformações à rigidez do material, englobando propriedades como módulo de elasticidade, coeficiente de expansão térmica e,

em modelos mais complexos, comportamento plástico ou viscoplástico. Por fim, F pode incluir forças externas, forças de volume (por exemplo, gravidade) e carregamentos térmicos decorrentes de gradientes de temperatura. (Zienkiewicz; Taylor; Zhu, 2005)

2.3 REDES NEURAS

As redes neurais artificiais são modelos computacionais inspirados na estrutura do cérebro humano, projetados para simular a forma como os neurônios biológicos processam e transmitem informações. Essas redes têm a capacidade de aprender padrões e relações complexas a partir de dados, sendo amplamente utilizadas em diversas aplicações, como reconhecimento de imagens, processamento de linguagem natural e análise de dados tridimensionais. (Goodfellow; Bengio; Courville, 2016)

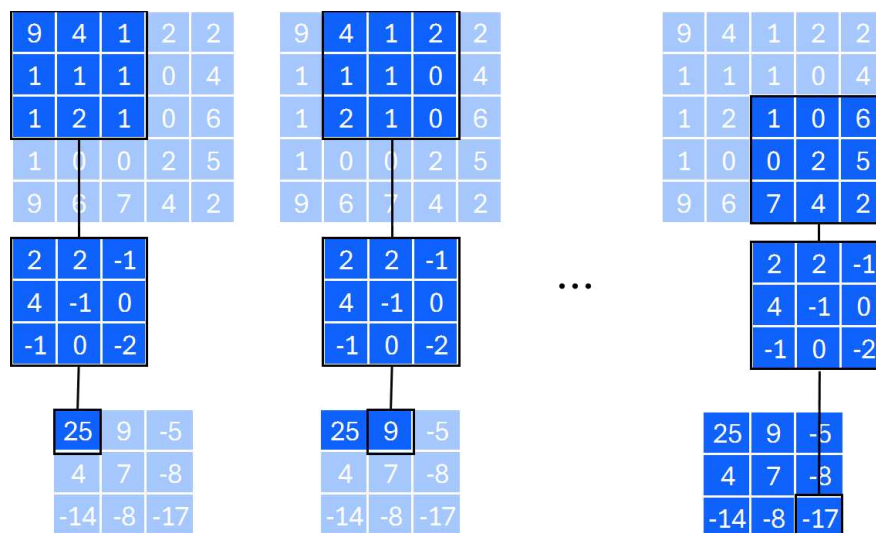
2.3.1 Redes Neurais Convolucionais

As camadas convolucionais desempenham um papel fundamental no processamento de dados de alta dimensionalidade, como imagens e volumes tridimensionais. Essas camadas têm a capacidade de reconhecer padrões espaciais e hierarquias de características, permitindo que a rede capture informações locais e globais presentes nos dados. A operação básica realizada nas camadas convolucionais é a convolução, na qual filtros (também chamados de kernels) são aplicados aos dados de entrada para extrair características como bordas e formas. (Géron, 2019)

Os filtros das camadas convolucionais são pequenos tensores de quarta ordem para operações bidimensionais e quinta ordem para tridimensionais, compostos por números, também chamados de pesos, que são ajustados durante o treinamento da rede neural. O objetivo dos pesos é transformar a imagem de entrada, através da operação de convolução, expondo características específicas que sejam relevantes para a rede neural. (Goodfellow; Bengio; Courville, 2016)

A operação de convolução, ilustrada na Figura 5, é executada da seguinte forma usando um filtro bidimensional de dimensão 3x3: a imagem de entrada, representada pela matriz 5x5, tem sua primeira submatriz 3x3 (situada no canto superior esquerdo) multiplicada termo a termo pelo filtro, representado pela matriz 3x3. Após a multiplicação, todos os produtos são somados e o resultado é posicionado na primeira posição de uma nova matriz, chamada de mapa de características. Em seguida, a operação se repete na submatriz situada uma posição à direita da mesma forma - multiplica-se termo a termo com o filtro e os produtos são somados, e o resultado é posicionado agora na segunda posição do mapa de características. A operação se repete até a última submatriz, situada no canto inferior direito da imagem de entrada. (Goodfellow; Bengio; Courville, 2016)

Figura 5 – Ilustração da operação de convolução



Fonte: Elaborada pelo autor (2025).

Uma camada convolucional normalmente possui vários filtros, e, portanto, vários mapas de características são gerados e várias características são extraídas da imagem de entrada simultaneamente. Dessa forma, cada filtro dentro da camada convolucional se especializa em capturar um tipo distinto de padrão na imagem de entrada, como bordas horizontais, verticais, curvas ou texturas mais complexas. (Géron, 2019)

Os mapas de características gerados são submetidos em seguida a uma função de ativação, o que será abordado na próxima subseção. Porém após a

função de ativação ser aplicada, todos os mapas de características gerados são utilizados como entrada da próxima camada convolucional, possibilitando a rede neural extrair outras características a partir daquelas já identificadas pelas camadas anteriores. (Goodfellow; Bengio; Courville, 2016)

A combinação desses mapas de características permite que a rede neural construa uma representação hierárquica dos dados, onde camadas iniciais identificam estruturas mais simples e camadas mais profundas aprendem abstrações progressivamente mais sofisticadas. Além disso, como diferentes filtros operam em paralelo, a rede adquire robustez na detecção de padrões mesmo diante de variações na escala, rotação ou iluminação da imagem. (Ronneberger; Fischer; Brox, 2015)

2.3.2 Funções de ativação

Para introduzir não linearidade no modelo e permitir que a rede aprenda relações complexas, utilizam-se funções de ativação que transformam a saída linear dos neurônios em uma representação mais rica dos dados (Goodfellow; Bengio; Courville, 2016). A seguir, discute-se em maior detalhe três das principais funções adotadas neste trabalho.

2.3.2.1 ReLU (*Rectified Linear Unit*)

A ReLU é definida pela equação (11), a qual ativa o resultado apenas quando a entrada é positiva. Essa característica não só acelera a convergência durante o treinamento, mas também ajuda a mitigar o problema do desvanecimento do gradiente, comum em redes profundas. (Goodfellow; Bengio; Courville, 2016)

$$f(x) = \max(0, x) \quad (11)$$

onde:

- $f(x)$ é o número resultante da ativação, que irá compor o mapa de características ativado
- x é o elemento do mapa de características “bruto”

Graficamente, a ReLU apresenta uma transição abrupta em zero, sendo linear para valores positivos e nula para valores negativos. Embora seja eficiente e computacionalmente simples, a ReLU pode levar ao fenômeno dos “neurônios mortos”, onde unidades que chegam a ter saída zero em uma iteração podem permanecer inativas nas subseqüentes, prejudicando o aprendizado. (Goodfellow; Bengio; Courville, 2016)

Quando aplicados a redes neurais em forma de U (*U-nets*), tema que será abordado na seção 2.3.7 - Arquiteturas U-net, ou arquiteturas autoencoders, a função de ativação ReLU é utilizada somente na parte de decodificação da rede neural, onde a probabilidade de ocorrer o problema dos neurônios mortos é maior. (Pintelas; Livieris; Pintelas, 2021)

2.3.2.2 Leaky ReLU

Para contornar a limitação dos neurônios mortos, a Leaky ReLU foi proposta. Sua formulação é dada pela equação (12), onde o coeficiente α (geralmente em torno de 0.01) que permite um gradiente pequeno mesmo para entradas negativas. (Goodfellow; Bengio; Courville, 2016)

$$f(x) = \max(\alpha x, x) \quad (12)$$

onde:

- $f(x)$ é o número resultante da ativação, que irá compor o mapa de características ativado
- x é o elemento do mapa de características “bruto”
- α é um coeficiente

Esse ajuste evita que o mapa de características se torne totalmente inativo e mantém uma pequena variação na região negativa, promovendo uma aprendizagem mais robusta, especialmente em arquiteturas profundas. (Goodfellow; Bengio; Courville, 2016)

2.3.2.3 Sigmoide

A função Sigmoide é definida pela equação, mapeando qualquer valor de entrada para o intervalo (0, 1). Essa propriedade a torna particularmente útil para problemas de classificação binária, onde a saída pode ser interpretada como uma probabilidade. (Géron, 2019)

$$f(x) = \frac{1}{1 + e^{-x}} \quad (13)$$

onde:

- $f(x)$ é o número resultante da ativação, que irá compor o mapa de características ativado
- x é o elemento do mapa de características “bruto”

No caso de processamento de geometrias por redes neurais, a função sigmoide é frequentemente utilizada somente na última camada da rede neural, a fim de inferir a cada voxel a probabilidade de existir material naquela posição. Dessa forma, os voxels cujos valores são mapeados para zero ou quase zero pela rede neural, são interpretados como espaços vazios na geometria e os voxels cujo valor são mapeados para um ou quase um, são interpretados como região com material. (Zhao *et al.*, 2019)

No entanto, a Sigmoide apresenta desvantagens em redes profundas, pois suas derivadas tendem a ser muito pequenas para valores de entrada distantes de zero, ocasionando o problema do desvanecimento do gradiente. (Goodfellow; Bengio; Courville, 2016)

2.3.3 Treinamento da rede neural

O treinamento da rede neural envolve o ajuste dos pesos que compõem os filtros de cada camada convolucional, de modo a minimizar a diferença entre as previsões da rede e os valores reais. Esse processo comumente utiliza o método de descida do gradiente aplicado à função de erro, que fornece a direção e a magnitude em que os pesos devem ser ajustados para reduzir o erro. (Shelhamer; Long; Darrell, 2016)

2.3.3.1 Algoritmos de otimização

Algoritmos de otimização são métodos matemáticos destinados a ajustar os parâmetros de um modelo (por exemplo, os pesos em uma rede neural) de modo a minimizar (ou maximizar) uma função objetivo, frequentemente chamada de função de custo ou perda. A equação (14) mostra a forma geral desta operação. (Géron, 2019)

$$\theta^* = \arg \min_{\theta} J(\theta) \quad (14)$$

onde:

- $J(\theta)$ é a função de custo
- θ^* é o argumento ótimo que minimiza o custo

Para alcançar esse objetivo, os algoritmos de otimização iteram sobre os parâmetros, atualizando-os de forma a diminuir $J(\theta)$. A abordagem clássica é o gradiente descendente, que se baseia na ideia de que a direção do gradiente $\nabla_{\theta} J(\theta)$ indica a taxa de variação da função em relação a cada parâmetro. Assim, uma atualização simples é feita conforme a equação (15). (Goodfellow; Bengio; Courville, 2016)

$$\theta_{t+1} = \theta_t - \alpha \nabla_{\theta} J(\theta_t) \quad (15)$$

onde:

- θ_t é o vetor de parâmetros na iteração t ,
- α é a taxa de aprendizado,
- $\nabla_{\theta} J(\theta_t)$ é o gradiente da função de custo avaliado em θ_t .

A ideia é mover θ na direção oposta ao gradiente, que aponta para o aumento da função, de forma a encontrar um mínimo.

2.3.3.1.1 Otimizador AdamW

O otimizador AdamW (*Adaptive Moment Estimation with decoupled Weight decay*) é uma variante aprimorada do algoritmo Adam tradicional. Assim como o Adam, ele combina os benefícios do Momentum, que suaviza a trajetória de atualização dos parâmetros, e do RMSprop, que adapta a taxa de

aprendizado com base na variância recente dos gradientes. No entanto, o AdamW corrige uma limitação conceitual importante do Adam original: o acoplamento inadequado entre a regularização L2 (também chamada de *weight decay*) e o cálculo dos gradientes. (Géron, 2019)

No AdamW, o termo de regularização é desacoplado da estimativa do gradiente, sendo aplicado diretamente aos pesos antes da atualização. Essa abordagem, descrita por *Loshchilov & Hutter (2017)* e também abordada por Géron (2019), melhora a generalização e proporciona um controle mais direto sobre a penalização dos parâmetros, especialmente em arquiteturas profundas.

Matematicamente, o AdamW preserva a estrutura original de estimativa dos momentos do otimizador Adam, com as seguintes equações:

1. Primeiro Momento:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} J(\theta_t) \quad (16)$$

onde:

- m_t é a média exponencialmente ponderada dos gradientes
- β_1 é um hiperparâmetro que controla o decaimento da média dos gradientes (primeiro momento)
- $\nabla_{\theta} J(\theta_t)$ é o gradiente da função de custo em relação aos parâmetros, avaliado na iteração t .

2. Segundo Momento:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) [\nabla_{\theta} J(\theta_t)]^2 \quad (17)$$

onde:

- v_t é a variável que acumula uma média exponencial ponderada dos quadrados dos gradientes anteriores,
- β_2 é um hiperparâmetro que controla o decaimento da média dos quadrados dos gradientes (segundo momento)

3. Atualização dos Parâmetros:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} - \alpha \cdot \lambda \cdot \theta_t \quad (18)$$

onde:

- ϵ termo de estabilidade numérica, adicionado para evitar divisão por zero (geralmente um valor muito pequeno, como 10^{-8}).
- α é a taxa de aprendizado,
- λ é o fator de *weight decay*,
- \hat{m}_t e \hat{v}_t são os momentos corrigidos para viés.

O termo $-\alpha \cdot \lambda \cdot \theta_t$ representa a penalização direta sobre os pesos do modelo, aplicada de forma desacoplada do gradiente. Essa estratégia, característica do AdamW, promove regularização explícita, ajudando a evitar sobreajuste sem interferir na direção do gradiente da função de custo. (Loshchilov; Hutter, 2017)

2.3.3.2 Função de erro BCE

A função de erro *Binary Crossentropy* é amplamente utilizada em tarefas de classificação binária, medindo a discrepância entre as previsões da rede e os valores reais. Ela é definida pela equação (19). (Dablain *et al.*, 2022)

$$BCE = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (19)$$

Onde:

- N representa o número de amostras
- y_i é o valor real da i -ésima amostra (valor esperado)
- \hat{y}_i é a previsão do modelo para a i -ésima amostra

Além de quantificar o erro das previsões para fins estatísticos, a curva de erro gerada pela equação (19) é utilizada pelo otimizador, através do gradiente, para encontrar a direção necessária para ajustar os pesos, a fim de que a resposta da rede neural seja mais próxima do valor real. Por esse motivo, as equações de erro devem ser diferenciáveis em todos os pontos para que o gradiente seja contínuo. Um ponto positivo da função de erro BCE é que

o erro calculado é calculado em escala logarítmica, tornando o gradiente mais inclinado em estágios iniciais onde as discrepâncias entre o real e o predito são maiores. (Chollet, 2021)

Figura 6 – Exemplo de cálculo do erro BCE

	Real (y)					Previsto (\hat{y})					Erro (BCE)				
	0	0	0	0	0	0,02	0,37	0,38	0,40	0,41	-0,01	-0,20	-0,21	-0,22	-0,23
	0	0	1	1	1	0,02	0,38	0,45	0,47	0,48	-0,01	-0,21	-0,35	-0,33	-0,32
	0	0	1	1	1	0,10	0,40	0,49	0,59	0,61	-0,05	-0,22	-0,31	-0,23	-0,21
	0	0	1	1	1	0,15	0,45	0,56	0,72	0,75	-0,07	-0,26	-0,25	-0,14	-0,12
	0	0	1	1	1	0,46	0,53	0,63	0,86	0,88	-0,27	-0,33	-0,20	-0,07	-0,06
	0	0	1	1	1	0,52	0,59	0,69	0,87	0,87	-0,32	-0,39	-0,16	-0,06	-0,06

Fonte: Elaborada pelo autor (2025).

O erro calculado pela função de erro BCE é executado conforme exemplifica a Figura 6. A matriz calculada pela rede neural, que contém os valores preditos (\hat{y}), é comparada termo a termo com a matriz que contém os valores reais (y). A equação do erro BCE calcula então um valor de erro para cada elemento da matriz com base na distância do mesmo até a resposta esperada. Isso faz com que o erro seja intensificado em escala logarítmica, ou seja, erros maiores que 0,5, por exemplo, são severamente penalizados enquanto erros menores que 0,5 são mais brandos. Em seguida a equação efetua uma média dos valores de erro de toda a matriz, para contabilizar um erro total da previsão. (Goodfellow; Bengio; Courville, 2016; TensorFlow, 2024a)

2.3.4 Técnicas de regularização

Durante o treinamento, as redes neurais estão sujeitas ao problema de sobreajuste, que é caracterizado pela contínua redução do erro medido no conjunto de treinamento e ausência de redução, ou aumento do erro no conjunto de validação. Ocorre devido a rede neural se ajustar excessivamente aos dados de treinamento, perdendo, assim, a capacidade de generalização. (Ertel, 2017)

Técnicas de regularização são fundamentais para mitigar o problema do sobreajuste em redes neurais, promovendo modelos com maior capacidade de generalização e uma abordagem amplamente utilizada é o Dropout. (Goodfellow; Bengio; Courville, 2016)

2.3.4.1 Dropout

O *Dropout* é uma técnica que, durante o treinamento, desativa aleatoriamente uma fração dos neurônios (ou filtros, no caso de camadas convolucionais) de uma camada em cada iteração. Essa desativação é controlada por um hiperparâmetro p , que define a probabilidade de um neurônio ser “desligado”. Ao impedir que os neurônios dependam excessivamente uns dos outros — um fenômeno conhecido como co-adaptação — o *Dropout* força a rede a aprender representações redundantes e robustas. Essa técnica tem se mostrado eficaz em diversas arquiteturas, reduzindo a complexidade do modelo sem comprometer sua capacidade de aprendizagem. (Warde-Farley *et al.*, 2013)

2.3.5 Normalização em lote

A normalização por lote, ou Batch Normalization, é uma técnica que visa estabilizar e acelerar o treinamento de redes neurais e é especialmente útil em redes neurais convolucionais, onde os mapas de características podem variar bastante em seus valores, impactando o desempenho do treinamento (Ioffe; Szegedy, 2015). A seguir, o funcionamento dessa técnica é descrito, focando nas particularidades dos mapas de características:

1. Coleta dos Mapas de Características do Mini-Lote: Em uma rede neural convolucional, cada camada convolucional gera vários mapas de características, e durante o treinamento, as imagens são processadas em grupos chamados mini-lotes. Para a normalização, todos os valores desses mapas, ou seja, todos os voxels de todas as imagens do mini-lote, são considerados juntos. (Goodfellow; Bengio; Courville, 2016)
2. Cálculo das Estatísticas do Mini-Lote: São calculadas a média e o desvio padrão de todos os valores presentes nos mapas de características do

- mini-lote Esses cálculos fornecem a distribuição dos valores que serão normalizados. (Ioffe; Szegedy, 2015)
3. Normalização das Ativações: Cada valor da previsão é ajustado subtraindo a média e dividindo pelo desvio padrão, transformando os dados para uma distribuição com média zero e variância unitária. Esse passo garante que os valores sejam padronizados, reduzindo variações excessivas entre as ativações. (Goodfellow; Bengio; Courville, 2016)
 4. Aplicação de Parâmetros de Escala e Deslocamento: Após a normalização, para não limitar a capacidade do modelo de representar dados complexos, a normalização é seguida por uma transformação linear: cada ativação normalizada é multiplicada por um parâmetro de escala (γ) e então somada a um parâmetro de deslocamento (β). Esses parâmetros são aprendidos durante o treinamento, permitindo que a rede recupere a flexibilidade na representação dos dados, caso a normalização “padrão” não seja ideal para determinada tarefa. (TensorFlow, 2024b)
 5. Integração no Fluxo de Dados da Rede: Esse processo é aplicado a cada mini-lote, em cada canal dos mapas de características gerados por uma camada convolucional. Ao garantir que, em cada mini-batch, as ativações estejam dentro de uma faixa estável, o Batch Normalization facilita o treinamento das camadas subsequentes. Essa estabilidade ajuda a reduzir a sensibilidade à escolha inicial dos pesos e à taxa de aprendizado, contribuindo para uma convergência mais rápida e eficaz. (Goodfellow; Bengio; Courville, 2016)

Em resumo, o Batch Normalization, ao padronizar os mapas de características em cada mini-lote e ao aplicar ajustes treináveis, torna o treinamento de redes neurais convolucionais mais robusto e eficiente, proporcionando um fluxo de dados mais uniforme e controlado através das camadas. (Zhang *et al.*, 2019)

2.3.6 Técnicas de redução e ampliação dimensional

As redes neurais U-Net empregam técnicas de redução e ampliação dimensional para construir uma arquitetura encoder-decoder que integra

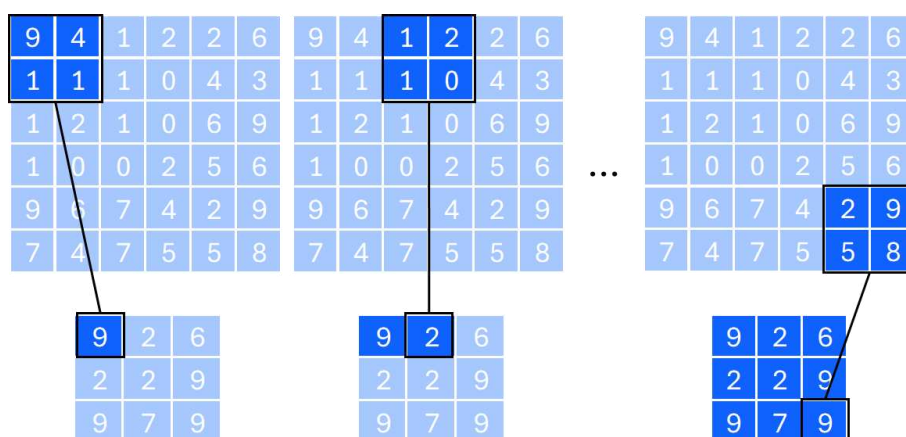
informações globais e locais através da redução dimensional dos exemplos de treinamento. (Arafin; Billah; Issa, 2024)

Durante a fase de codificação, o uso de técnicas como o *Max Pooling* permite a compactação dos dados, sintetizando informações contextuais e executando o reconhecimento de padrões (Çiçek *et al.*, 2016). Em contrapartida, na fase de decodificação pode ser utilizada a Convolução Transposta para ampliar progressivamente os mapas de características, restaurando a resolução original e possibilitando a reconstrução detalhada das representações. Essa combinação garante que a U-Net mantenha tanto a compreensão contextual quanto a precisão espacial, elementos fundamentais para o sucesso em tarefas de segmentação e análise de imagens complexas. (Ronneberger; Fischer; Brox, 2015)

2.3.6.1 Max pooling

O Max Pooling atua na redução da dimensão ao condensar informações locais relevantes. Nesta técnica, define-se uma janela de tamanho fixo — por exemplo, 2x2 para imagens 2D ou 2x2x2 para volumes 3D — que varre a entrada de forma não sobreposta ou com sobreposição, conforme a aplicação. Em cada janela, o algoritmo calcula a média dos valores dos pixels (ou voxels), que é então armazenada em uma posição correspondente no mapa de características reduzido. (Géron, 2019)

Figura 7 – Exemplo de aplicação do *Max Pooling*



Fonte: Elaborada pelo autor (2025).

A Figura 7 mostra um exemplo de aplicação da operação *Max Pooling* em uma matriz de dimensão 6x6. Na primeira etapa do procedimento, o programa seleciona a primeira submatriz de dimensão 2x2 (considerando uma janela de dimensão 2x2, neste caso), e calcula a média dos valores e a posiciona na primeira posição da matriz resultante. O mesmo é feito para a segunda submatriz localizada duas posições à direita, dado que o passo utilizado nesta operação é igual a 2. Esse procedimento é repetido até a última submatriz localizada no canto inferior direito da matriz de entrada. Dessa forma, a dimensão da matriz resultante é metade da dimensão da matriz de entrada, o que reduz em 4 vezes o número de elementos, considerando dados bidimensionais, e portanto, 4 vezes menos operações em alguns recursos computacionais. (Çiçek *et al.*, 2016; Nasiri *et al.*, 2020)

Essa operação não só reduz o número de parâmetros e o custo computacional, como também confere aos filtros da rede neural maior alcance dentro da geometria, ou seja, a rede consegue enxergar uma parcela maior da geometria. Isso é especialmente relevante para camadas convolucionais que utilizam filtros de baixa dimensão quando comparados a dimensão da imagem processada pela mesma. (Goodfellow; Bengio; Courville, 2016)

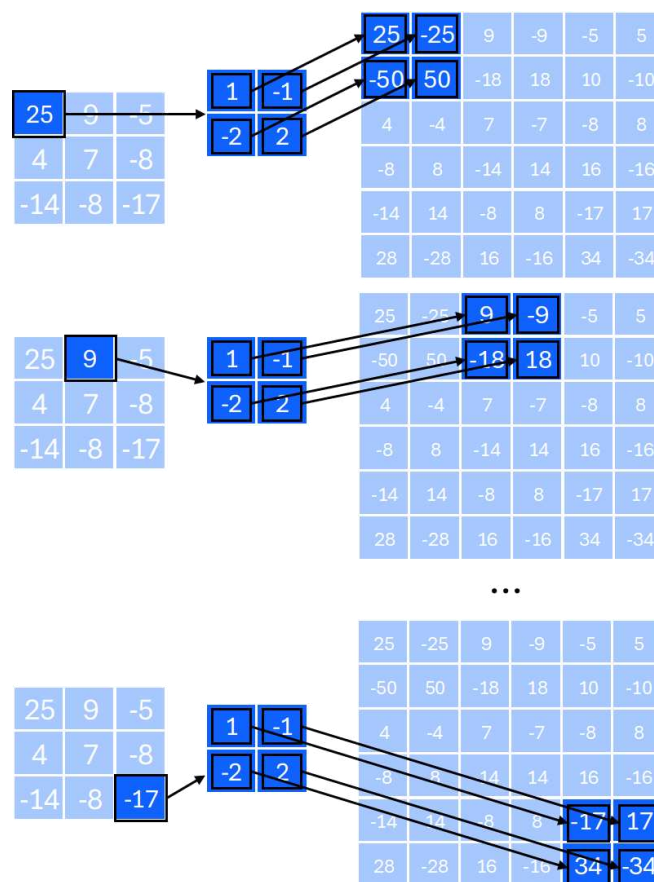
2.3.6.2 Convolução transposta

No outro lado da *U-Net*, na etapa de decodificação, a Convolução Transposta pode ser utilizada para ampliar a dimensão dos dados, sendo especialmente importante em tarefas que exigem reconstrução ou geração de imagens de alta resolução, como na segmentação semântica e em arquiteturas de *autoencoders*. (Shen *et al.*, 2019)

Diferente da convolução tradicional, que comprime a informação, a convolução transposta realiza uma operação inversa: ela “espalha” os valores de um mapa de características menor por meio de um filtro treinável, expandindo-o para gerar uma saída com dimensões maiores (Nasiri *et al.*, 2020). A Figura 8 mostra um exemplo de cálculo de convolução transposta aplicada a uma matriz de entrada com dimensão 3x3. O filtro utilizado neste exemplo é de dimensão 2x2, e, a fim de duplicar a dimensão dos dados de entrada, o passo da operação foi considerado igual a 2.

No início da operação, o primeiro elemento da matriz de entrada é multiplicado pelos quatro elementos do filtro, e os resultados são posicionadas na primeira submatriz de dimensão 2x2 da matriz resultante. A próxima etapa da operação é multiplicar o segundo elemento da matriz de entrada pelos quatro elementos do filtro, e o resultado é posicionado na submatriz localizada a duas posições à direita da primeira submatriz, visto que o passo da operação é igual a 2. A operação se repete até o último elemento da matriz de entrada, situado no canto inferior direito. (Goodfellow; Bengio; Courville, 2016)

Figura 8 – Exemplo de operação de convolução transposta



Fonte: Elaborada pelo autor (2025).

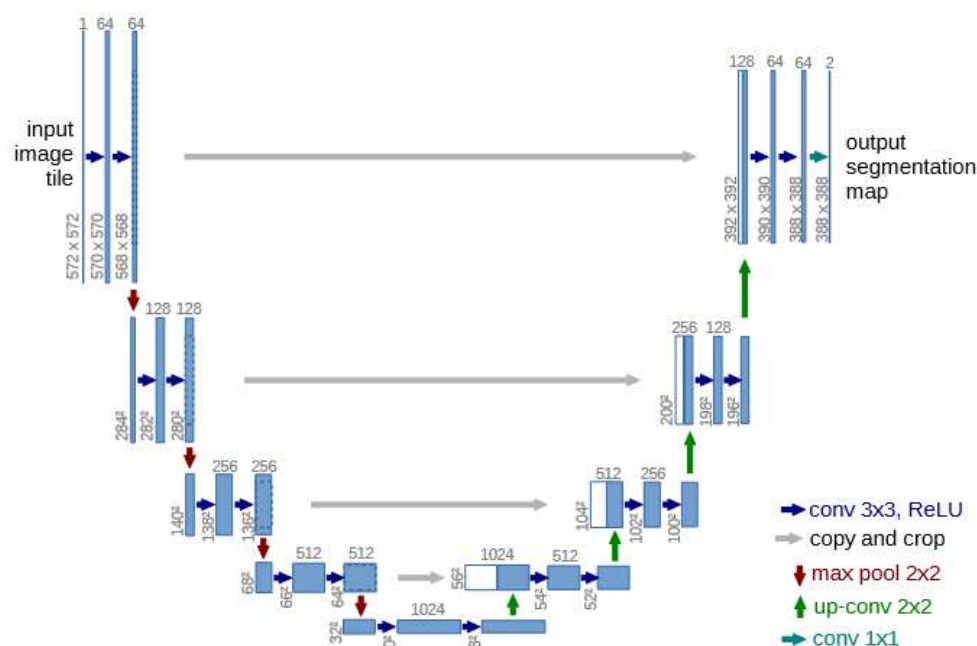
Uma abordagem alternativa a convolução transposta envolve inserir zeros entre os elementos do mapa de entrada — processo denominado “*up-sampling*” — seguido de uma convolução convencional, cujo filtro é ajustado durante o treinamento para corrigir as posições preenchida com zeros, substituindo-as por valores preditos pela rede neural. (Géron, 2019)

2.3.7 Arquiteturas U-net

A utilização das redes neurais convolucionais em arquiteturas mais especializadas, como a *U-net*, as tornam capazes de lidar com tarefas específicas, como a segmentação semântica de imagens. A arquitetura *U-net*, que se baseia nos princípios das redes neurais convolucionais, é especialmente projetada para extrair e utilizar contextos espaciais em detalhes finos, o que a torna particularmente eficaz para a segmentação de imagens médicas, por exemplo, onde precisão e detalhe são imprescindíveis. (Ronneberger; Fischer; Brox, 2015)

O *design* da *U-net* é caracterizado por um caminho de contração e um caminho de expansão. No caminho de contração, a arquitetura segue uma estrutura típica de uma rede convolucional, aplicando repetidamente convoluções seguidas de operações de *pooling* para reduzir a dimensão dos dados de entrada. Esse processo força a rede neural a priorizar apenas os aspectos úteis do conjunto de dados e possibilita a compreensão de aspectos mais profundos a respeito da imagem. A redução da dimensão dos dados de entrada também reduz a quantidade de parâmetros da rede, o que, por sua vez, melhora a eficiência computacional. (Zhao *et al.*, 2019)

Figura 9 – Arquitetura de uma U-net



Fonte: (Ronneberger; Fischer; Brox, 2015)

Já o caminho de expansão da *U-net* visa reconstruir a resolução espacial dos dados de saída para corresponder ao tamanho original da imagem de entrada. Isso é realizado através de operações de *upsampling* e convoluções, que gradualmente aumentam a dimensionalidade espacial dos mapas de características. A ideia é restaurar os detalhes e a localização exata dos objetos de interesse na imagem, o que é essencial para tarefas de segmentação precisa. (Zhao *et al.*, 2019)

Um componente importante que diferencia a *U-net* de outras arquiteturas convolucionais é a implementação de conexões diretas, conhecidas como *skip connections*, entre camadas de mesma dimensão no caminho de contração e no caminho de expansão, conforme representado na Figura 9 por setas horizontais. Essas conexões permitem que a rede reintroduza os detalhes espaciais perdidos durante a contração diretamente para as camadas de *upsampling*, mitigando o problema do erro de informação espacial que pode ocorrer durante a contração. Sem as *skip connections*, a rede pode ter dificuldade em reconstruir detalhes finos nos dados de saída, resultando em segmentações menos precisas e bordas difusas. (Shen *et al.*, 2019)

2.3.8 Redes convolucionais para dados tridimensionais

Em tarefas de segmentação semântica de imagens, redes neurais convolucionais (CNNs) vêm sendo empregadas devido à sua capacidade de aprender representações hierárquicas a partir dos dados. Nesse contexto, Shelhamer *et al.* (2015) propuseram e caracterizaram as Redes Totalmente Convolucionais (*Fully Convolutional Networks* – FCNs), adaptando arquiteturas de classificação como AlexNet, VGG e GoogLeNet para gerar mapas de segmentação detalhados a partir de imagens de entrada de tamanho arbitrário. Ao converter arquiteturas de classificação de imagens já consolidadas e ajustá-las à tarefa de segmentação, seu trabalho demonstrou que CNNs podem ser treinadas de ponta a ponta, dispensando etapas complexas de pré e pós-processamento. Além disso, a introdução de uma arquitetura com *skip connections* permitiu combinar informações semânticas de camadas mais profundas com detalhes das camadas mais rasas, resultando em segmentações mais precisas e detalhadas. (Shelhamer; Long; Darrell, 2016)

Posteriormente, Ronneberger et al. (2015) aprimoraram essa ideia ao introduzir a U-Net, uma arquitetura específica para segmentação semântica de imagens biomédicas. A U-Net caracteriza-se por sua estrutura em forma de “U”, composta por um caminho de contração (*encoder*) e um caminho de expansão (*decoder*), bem como pelo uso sistemático de conexões que transferem informações entre camadas equivalentes dos dois caminhos. Esse desenho arquitetônico facilita a recuperação de detalhes espaciais, o que contribui para a segmentação de estruturas finas ou complexas. No trabalho original, Ronneberger et al. aplicaram a U-Net para segmentação de estruturas celulares em imagens de microscopia, obtendo resultados expressivos. Eles demonstraram a eficácia da U-Net na detecção e segmentação acurada de células e tecidos em imagens médicas, mesmo com um número limitado de dados anotados, algo comum na área médica devido à dificuldade de obtenção de grandes conjuntos de dados rotulados. (Ronneberger; Fischer; Brox, 2015)

Reconhecendo a necessidade de processar dados volumétricos em aplicações médicas, Çiçek et al. (2016) estenderam a U-Net para dados tridimensionais, criando a 3D U-Net. Essa arquitetura adapta as convoluções e operações de pooling para o domínio tridimensional, permitindo o processamento direto de volumes, como imagens de ressonância magnética ou tomografia computadorizada. No estudo, Çiçek et al. aplicaram a 3D U-Net para segmentação de estruturas internas em dados volumétricos, como o cérebro humano em imagens de ressonância magnética. Eles demonstraram que a 3D U-Net é capaz de aprender representações volumétricas densas a partir de anotações esparsas, alcançando segmentações precisas de órgãos e tecidos internos. Isso foi particularmente significativo para aplicações como planejamento cirúrgico e diagnóstico, onde a compreensão tridimensional detalhada da anatomia é importante. (Çiçek *et al.*, 2016)

A aplicação de arquiteturas baseadas em U-Net para dados 3D não se limita ao domínio médico. Wang et al. (2015) propuseram o CAE-ELM (*Convolutional Auto-Encoder Extreme Learning Machine*), uma abordagem que integra redes convolucionais, autoencoders e máquinas de aprendizado extremo, visando superar limitações de métodos tradicionais e outras arquiteturas profundas. Utilizando dados voxelizados e campos de distância

assinados (SDF) para representar tanto a estrutura quanto detalhes geométricos, o CAE-ELM alcançou elevadas taxas de classificação (91,41% no ModelNet10 e 84,35% no ModelNet40), superando abordagens anteriores e reduzindo significativamente o tempo de treinamento (da ordem de duas magnitudes mais rápido). Além disso, a rede demonstrou desempenho robusto em tarefas de recuperação de formas e completude de objetos 3D danificados, bem como bom desempenho em dados bidimensionais (MNIST e Norb), evidenciando sua versatilidade e eficiência no aprendizado de características tridimensionais. (Wang *et al.*, 2016)

No contexto de processos de fabricação, alguns processos provocam desvios dimensionais nas peças que tornam necessário aplicar compensações dimensionais previamente à geometria de entrada. Zhao *et al.* (2019) propuseram uma abordagem baseada em autoencoders convolucionais (CAE) para prever e corrigir deformações não lineares, com foco no processo de impressão 3D por processamento digital de luz (DLP). O trabalho consiste em dois métodos: na primeira, a rede neural faz a previsão da geometria deformada a partir da geometria inicial, a fim de determinar qual será a geometria de uma peça após sua impressão 3D considerando as deformações provocadas durante o processo; na segunda, uma rede neural prediz a geometria inicial necessária para que a geometria final seja mais parecida com a geometria nominal, ou seja, a rede neural é treinada para calcular as compensações dimensionais que devem ser aplicadas à geometria de entrada. Os dados são representados como grades de voxels 3D, e o treinamento considera tanto deformações lineares quanto não lineares. Resultados quantitativos, expressos por métricas como F1-score, mostram que a rede alcança alta capacidade de generalização para diferentes resoluções (32^3 e 64^3), aproximando deformações complexas com precisão próxima ou superior a 0.95 de F1. Embora as resoluções utilizadas ainda sejam inferiores às exigidas em aplicações práticas, o estudo demonstra o potencial das redes neurais *Auto-Encoders* para lidar com erros dimensionais em geometrias tridimensionais. (Zhao *et al.*, 2019)

De forma semelhante às abordagens baseadas em CAE propostas por Zhao (2019), Shen *et al.* (2019) desenvolveram uma proposta sistêmica de aprendizado profundo para compensação de distorções em impressão 3D, desta

vez utilizando uma rede de função inversa estruturada sobre a arquitetura U-Net. A solução recebe como entrada modelos impressos deformados e, ao processá-los, gera modelos compensados prontos para impressão, abordando tanto deformações simples (como translação e escalonamento) quanto não lineares. Em um conjunto abrangente de dados — 39.424 modelos voxelizados de coroas dentárias — a rede demonstrou resultados comparáveis aos de Zhao em termos de precisão ($F1 > 0,93$), porém com uma abordagem mais abrangente. A integração da abordagem a um sistema ciber-habilitado (CeMS) permite conectar sensoriamento, computação e controle, criando um loop fechado de *feedback* que viabiliza ajustes mais robustos e generalizáveis. Dessa forma, Shen et al. (2019) expandem o escopo da compensação de deformações, reforçando o potencial de arquiteturas profundas em lidar com os desafios de precisão na manufatura aditiva. (Shen *et al.*, 2019)

É importante notar que o processamento de dados 3D apresenta desafios computacionais significativos. Zhang et al. (2019) abordaram o gerenciamento de memória em sistemas de aprendizado profundo baseados em GPU propondo o *SmartPool* e o *AutoSwap*, mecanismos que otimizam a alocação de memória e o intercâmbio de variáveis entre GPU e CPU. O *SmartPool* reduz a fragmentação ao reutilizar memória conforme o tempo de vida das variáveis, enquanto o *AutoSwap* transfere automaticamente variáveis ociosas para a CPU, aliviando a carga na GPU sem prejudicar o desempenho. Em seus experimentos, essas abordagens reduziram o uso de memória em até um terço, sem comprometer a precisão ou aumentar o tempo de treinamento, superando métodos anteriores como *MXNet-memonger* e *SuperNeurons*. Essas estratégias são essenciais ao treinar arquiteturas tridimensionais complexas, como a U-Net 3D, que demandam grande capacidade computacional devido ao processamento intensivo de dados volumétricos. (Zhang *et al.*, 2019)

2.4 PROCESSAMENTO DE IMAGENS TRIDIMENSIONAIS

No trabalho com redes neurais que envolvem dados em três dimensões, observa-se com frequência a adoção de representações voxelizadas. Essas abordagens permitem o uso de estruturas discretizadas em 3D, facilitando operações como classificação, segmentação e extração de características internas de objetos. Em diversos domínios — por exemplo, em análises médicas ou em inspeções industriais —, é comum que a informação volumétrica seja organizada em formatos compatíveis com métodos de aprendizado profundo, incluindo arranjos de voxels. (Wang *et al.*, 2016)

2.4.1 Definição de Voxel

A palavra voxel tem sua origem na junção dos termos volume e pixel, pois designa o análogo tridimensional do pixel. Enquanto o pixel descreve a menor unidade de informação em uma imagem 2D, o voxel representa uma porção elementar de um volume em um espaço 3D. Assim, cada voxel pode ser entendido como um cubo (ou célula) de tamanho pré-definido que compõe uma malha regular distribuída em três dimensões. (Zhao *et al.*, 2019)

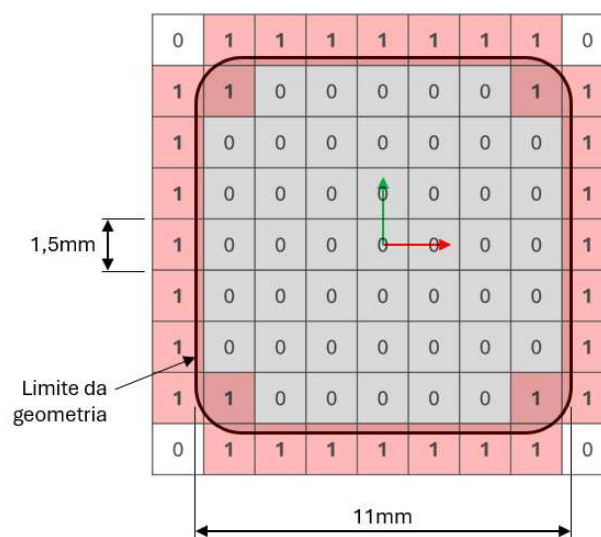
No contexto de computação gráfica, cada voxel pode conter informações de forma binária (ativo/inativo) ou valores mais complexos, como intensidades em escala de cinza, cores ou até parâmetros físicos (densidade, temperatura, etc.). Essa versatilidade possibilita amplo uso em aplicações que envolvem modelagem e análise volumétrica, como visualização médica, simulações numéricas e algoritmos de inteligência artificial. (Wang *et al.*, 2016)

2.4.2 Voxelização

Uma geometria contínua formada por uma malha (mesh) de vértices e faces pode ser discretizada e transformada em uma imagem através da técnica de voxelização. Nesse procedimento, a malha é convertida em uma grade tridimensional na qual cada célula corresponde a um voxel.

A biblioteca *Trimesh* possui uma ferramenta de voxelização denominada *Voxelize*, a qual executa essa voxelização de malhas tridimensionais no ambiente de programação Python. O funcionamento do comando *Voxelize* é ilustrado na Figura 10, a qual mostra uma geometria quadrada com cantos arredondados durante o processo de voxelização. O espaço que contém a geometria é discretizado numa grade, neste caso de 1,5mm de espaçamento. O voxel inicial é disposto na origem do sistema de coordenadas, e os voxels ao redor são adicionados a cada 1,5mm de distância nas direções X e Y, neste caso. O programa então avalia quais voxels cruzam com a malha que define a geometria, e altera os valores destes voxels para 1, sinalizando que há material naquela posição. (Trimesh, 2022)

Figura 10 - Aumento dimensional após a voxelização



Fonte: Elaborada pelo autor (2025).

A geometria voxelizada resultante é então semelhante a uma casca, composta por uma superfície onde antes havia a malha, e o interior vazio. Para aplicações onde o interior da geometria deve ser preenchido, a ferramenta *binary_fill_holes* da biblioteca *SciPy* pode ser utilizada. Este outro comando é capaz de identificar o contorno da geometria e preencher os voxels internos com valores unitários, respeitando furações e passagens que a peça eventualmente tenha.

Embora esse processo facilite diversas operações — como manipulação automatizada em redes neurais —, a definição do tamanho dos voxels deve conciliar a fidelidade geométrica com as limitações de memória e processamento. Em geral, a redução no tamanho do voxel aumenta a precisão com que a geometria é representada, mas multiplica o volume de dados, demandando maior capacidade computacional. Por outro lado, voxels maiores diminuem o custo computacional, porém podem acarretar perda de detalhes relevantes da geometria. (Zhao *et al.*, 2019)

Além do problema relacionado ao consumo de memória para manipular geometrias voxelizadas, ocorre também a alteração dimensional das geometrias voxelizadas, principalmente em casos em que são usados voxels grandes. Esse fenômeno ocorre devido à resolução limitada dos voxels e à forma como eles são considerados ativos ou inativos com base na interferência com a malha original. A peça tomada como exemplo na Figura 10 não tem a representação das dimensões reais da peça com precisão, visto que a dimensão original da peça é 11mm e a dimensão após a voxelização é de 9 voxels, o que resulta em 13,5mm. Devido a resolução da grade, os voxels nas extremidades podem ser considerados ativos mesmo que apenas uma pequena parte deles interfira com a peça. Isso resulta em uma representação voxelizada que excede as dimensões originais, fazendo com que a peça pareça maior do que realmente é. (Eiriksson, 2018)

2.4.3 Métricas de Desempenho na Classificação Binária de Voxels com Inteligência Artificial

Em redes neurais aplicadas à segmentação e análise de voxels, as previsões calculadas pela rede neural são comparadas, durante o treinamento, com os valores reais, ou a resposta esperada e a partir dessa comparação, métricas devem ser utilizadas para quantificar os erros e acertos, e assim medir a qualidade das previsões. (Taha; Hanbury, 2015)

As principais métricas utilizadas incluem exatidão (*accuracy*), precisão (*precision*), sensibilidade (*recall*) e *F1-score*, sendo estas fundamentais para avaliar a qualidade das previsões geradas pelo modelo. Além disso, a análise de

falsos positivos e falsos negativos permite uma compreensão detalhada dos erros cometidos pelo classificador e possibilitam entender o comportamento das previsões do modelo. (Taha; Hanbury, 2015)

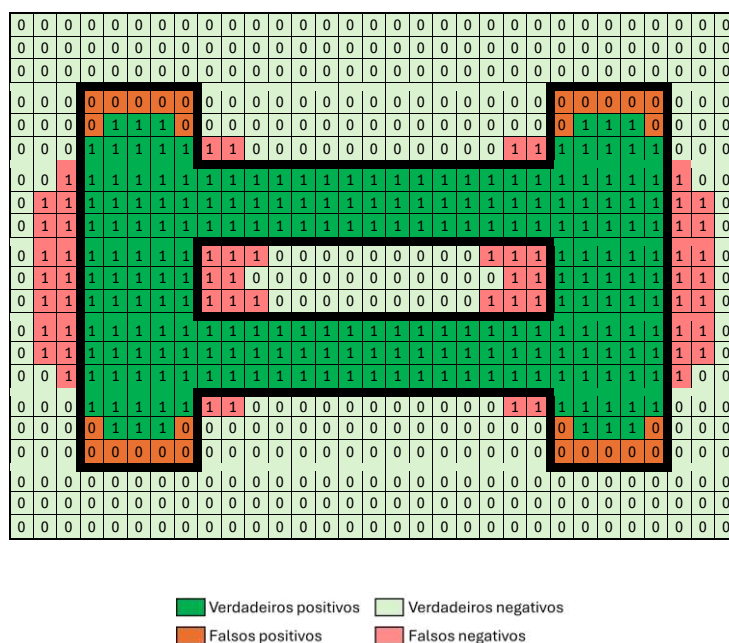
2.4.3.1 Definição de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos

A avaliação de modelos de redes neurais em tarefas de classificação é frequentemente realizada com o auxílio de métricas baseadas em quatro categorias de acerto e erro: verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN) (Géron, 2019). No contexto deste trabalho, em que se utilizam predições em forma de voxels para a reconstrução ou geração de uma peça, esses conceitos podem ser explicados da seguinte forma:

1. Verdadeiros Positivos (VP): São os voxels em que se esperava um valor unitário (1) e a rede neural previu o valor 1 (Géron, 2019). Em outras palavras, o voxel efetivamente deveria fazer parte da geometria da peça, e o modelo classificou-o corretamente como tal. Esses voxels formam a porção “sólida” da peça, ou seja, onde há material.
2. Verdadeiros Negativos (VN): Correspondem aos voxels em que se esperava um valor nulo (0) e a rede neural previu também o valor 0 (Géron, 2019). Isso significa que o voxel não faz parte da geometria e deve permanecer vazio, o que o modelo identificou corretamente. Esses voxels formam os “espaços vazios” ao redor ou dentro da peça.
3. Falsos Positivos (FP): Neste caso, o valor esperado para o voxel era nulo (0), porém a rede neural previu 1 (Géron, 2019). Ou seja, o modelo adiciona material em uma região que deveria ser vazia. Em aplicações reais, esses erros podem causar protuberâncias ou excesso de material em zonas onde não deveria haver material.
4. Falsos Negativos (FN): Aqui, o valor esperado para o voxel era 1, mas a rede neural previu 0 (Géron, 2019). Isso significa que o modelo remove material de uma área que, na verdade, precisaria ser sólida. O resultado prático é uma falta de material em alguma região da peça.

Essas quatro categorias compõem a chamada matriz de confusão na classificação binária. A análise conjunta desses valores permite avaliar o desempenho do modelo de forma mais precisa, identificando onde ocorrem os acertos (VP e VN) e os erros (FP e FN). No contexto de manufatura, impressão 3D ou outras aplicações em que a forma final do objeto é crítica, minimizar falsos positivos e falsos negativos é tão importante quanto maximizar verdadeiros positivos e verdadeiros negativos, pois ambos os tipos de erro afetam diretamente a forma do produto final. (Taha; Hanbury, 2015)

Figura 11 – ilustração dos verdadeiros e falsos positivos e negativos



Fonte: Elaborada pelo autor (2025).

A Figura 11 apresenta uma grade de voxels onde cada posição pode assumir valor 0 ou 1. A linha preta espessa contorna a geometria esperada (ou seja, a região na qual os voxels deveriam ser 1, pois compõem o objeto). Já a área fora desse contorno corresponde aos voxels que, em teoria, deveriam ser 0, representando espaços vazios.

Para facilitar a interpretação dos acertos e erros da rede neural ao prever essa configuração, foram utilizadas cores distintas em cada voxel:

- Verde escuro (Verdadeiros Positivos - VP): Voxels que deveriam ser 1 (dentro do contorno) e foram corretamente preditos como 1. Isso significa que a rede neural manteve material onde realmente deveria existir.
- Verde claro (Verdadeiros Negativos - VN): Voxels que deveriam ser 0 (fora do contorno) e foram corretamente preditos como 0, indicando que o modelo não preencheu material em regiões vazias.
- Vermelho (Falsos Positivos - FP): Voxels que deveriam ser 0, mas foram preditos como 1. Nesse caso, o algoritmo “adiciona” material indevidamente, criando uma protuberância fora da região esperada.
- Laranja (Falsos Negativos - FN): Voxels que deveriam ser 1, porém foram preditos como 0. Aqui, o modelo “remove” material onde deveria existir, gerando lacunas ou buracos na peça.

2.4.3.2 Exatidão

A exatidão é calculada pela equação (20)(7), obtendo-se o valor da divisão da contagem de acertos pelo número total de predições. Em outras palavras, é o valor da divisão da contagem de acertos pelo número total de voxels, ou volume total da imagem. (Taha; Hanbury, 2015)

$$E = \frac{VP + VN}{V_T} = \frac{VP + VN}{VP + VN + FP + FN} \quad (20)$$

onde:

- E é a exatidão
- VP é a contagem de verdadeiros positivos
- VN é a contagem de verdadeiros negativos
- FP é a contagem de falsos positivos
- FN é a contagem de falsos negativos
- V_T é o volume total da imagem ou o número total de previsões,

A Figura 12 mostra o exemplo usado na subseção anterior sob a ótica da exatidão, onde toda a área é considerada no cálculo. A área total da imagem é de 600 pixels, onde 84 deles foram preditos incorretamente pela rede neural, portanto, o número de acertos é de 516. Dessa forma a exatidão resultante é de 86%, visto que esse foi o percentual de acertos dentre as predições.

Figura 12 – Avaliação da previsão do ponto de vista da exatidão

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	
6	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
7	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
10	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
11	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
12	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
14	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
15	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborada pelo autor (2025).

O ponto negativo da exatidão é a limitação em conjuntos de dados desbalanceados. No caso de uma das classes ocupar a maior parte dos dados, a rede neural pode preencher toda a área da imagem com um único valor e ainda assim ter um alto percentual de acertos. Por este motivo, é recomendada a complementação dessa métrica com outras mais robustas, como precisão, sensibilidade ou *F1-score*, especialmente em aplicações que envolvem aprendizado profundo. (Chollet, 2021)

2.4.3.3 Precisão

A precisão mede a proporção de verdadeiros positivos em relação ao total de exemplos previstos como positivos, usando a equação (21). (Taha; Hanbury, 2015)

$$P = \frac{VP}{V_P} = \frac{VP}{VP + FP} \quad (21)$$

onde:

- P é a precisão
- VP é a contagem de voxels corretamente classificados como positivos
- FP é a contagem de voxels erroneamente classificados como positivos
- V_P é a contagem de voxels positivos

Essa métrica é particularmente importante em problemas onde falsos positivos têm um impacto significativo, como no diagnóstico de doenças em imagens médicas, onde um falso positivo é mais seguro que um falso negativo. No entanto, alta precisão não garante um bom desempenho geral do modelo, pois não considera os falsos negativos, que podem ser críticos dependendo da aplicação. (Taha; Hanbury, 2015)

Figura 13 - Avaliação da previsão do ponto de vista da precisão

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
7	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
10	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
11	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
12	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
14	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
15	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
16	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
17	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fonte: Elaborada pelo autor (2025).

A Figura 13 mostra um exemplo de cálculo de precisão na matriz de dados usada nos exemplos anteriores. Todos os elementos destacados possuem valor igual a 1, visto que somente estes são considerados no cálculo de precisão. O total de elementos positivos é de 262, e somente 206 foram calculados corretamente (verdadeiros positivos), a precisão neste caso é calculada em 78,6%

2.4.3.4 Sensibilidade

A sensibilidade, representa a fração de exemplos positivos corretamente identificados pelo modelo em relação ao total real de exemplos positivos, sendo calculada através da equação (22).

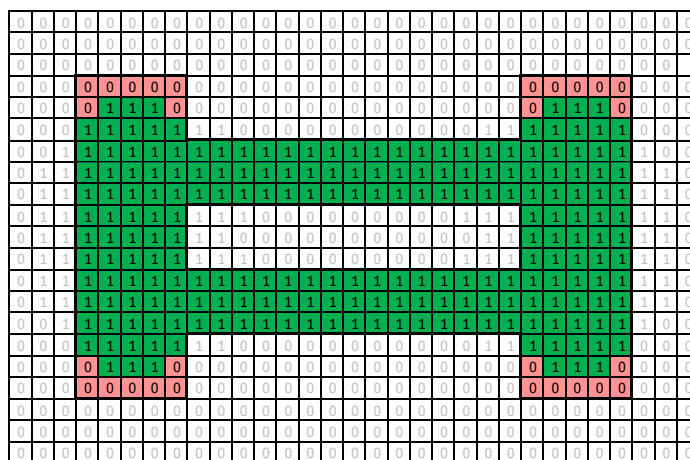
$$S = \frac{VP}{V_G} = \frac{VP}{VP + FN} \quad (22)$$

onde:

- S é a sensibilidade
- VP é a contagem de voxels corretamente classificados como positivos
- FN é a contagem de voxels erroneamente classificados como negativos
- V_G é o volume da geometria modelada pelos voxels

Essa métrica é essencial para aplicações onde a minimização de falsos negativos é prioritária, como em sistemas de segurança para minimização de “alarmes falsos”. Modelos treinados com a intenção de maximizar a sensibilidade tendem a ser mais permissivos, classificando um maior número de instâncias como positivas, mesmo que algumas dessas previsões sejam incorretas. (Chollet, 2021)

Figura 14 - Avaliação da previsão do ponto de vista da sensibilidade



Fonte: Elaborada pelo Autor (2025).

A Figura 14 mostra um exemplo de cálculo de sensibilidade. Todos os elementos da geometria esperada foram destacados, visto que somente estes são considerados no cálculo de sensibilidade. A área total da figura é de 234, e somente 206 foram calculados corretamente (verdadeiros positivos), a sensibilidade neste caso é calculada em 88%.

2.4.3.5 F1-Score

O F1-score é uma métrica que harmoniza precisão e recall, sendo calculado como a média harmônica dessas duas métricas conforme mostrado na equação (23). (Géron, 2019; Taha; Hanbury, 2015; Zhao *et al.*, 2019)

$$F1 = \frac{2}{P^{-1} + S^{-1}} = \frac{2 VP}{2 VP + FP + FN} \quad (23)$$

onde:

- F1 é o F1-score
- P é a precisão
- S é a sensibilidade
- VP é a contagem de voxels corretamente classificados como positivos
- VN é a contagem de voxels corretamente classificados como negativos
- FP é a contagem de voxels erroneamente classificados como positivos
- FN é a contagem de voxels erroneamente classificados como negativos

O seu uso é altamente indicado para aplicações em que os falsos negativos são igualmente importantes em relação aos falsos positivos. *O F1-score* se mostra particularmente útil na classificação de voxels em imagens tridimensionais, pois os falsos positivos, que agregam erroneamente material à geometria, modificam a geometria tanto quanto os falsos negativos, que removem material da peça erroneamente. (Taha; Hanbury, 2015)

Figura 15 – Avaliação da previsão do ponto de vista do F1-score

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
7	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
8	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
9	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
10	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
11	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
12	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0
13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
14	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
15	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
16	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
17	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

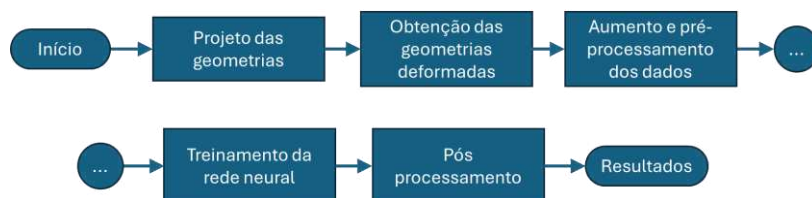
Fonte: Elaborada pelo Autor (2025).

A Figura 15 mostra um exemplo de cálculo de *F1-score*. Todos os elementos relevantes foram destacados, isto é, os elementos que compõem a geometria completa (VP + FN) mais os falsos positivos, visto que somente estes são considerados no cálculo de *F1-score*, ficando de fora apenas os verdadeiros negativos. Usando a equação (23) no exemplo mostrado na Figura 15, o valor de *F1-score* calculado é de 83,1%.

3 METODOLOGIA

O projeto consiste na elaboração de um programa que utilize redes neurais convolucionais para prever o comportamento dimensional de peças fundidas em areia. As etapas do projeto são mostradas no fluxograma da Figura 16, iniciando pelo projeto das geometrias utilizadas para treinar e avaliar a rede neural, seguido pela obtenção das geometrias deformadas através da simulação computacional e a etapa da multiplicação do número das geometrias. Essas geometrias processadas são usadas no treinamento da rede neural, que é a etapa na sequência. A performance da rede neural foi avaliada em seguida na etapa de pós processamento, onde são gerados os resultados.

Figura 16 – Fluxograma do projeto



Fonte: Elaborada pelo Autor (2025).


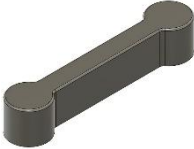
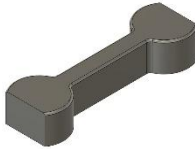
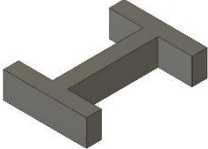
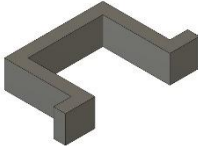
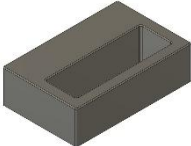
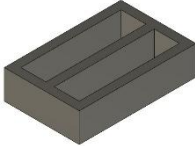
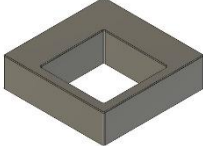
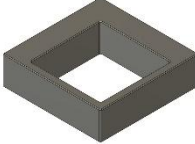
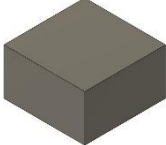
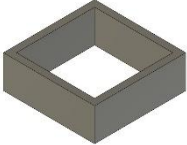
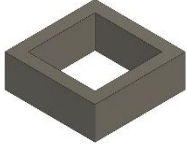
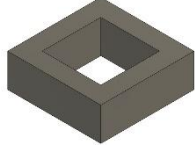

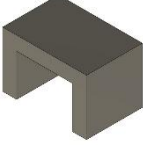
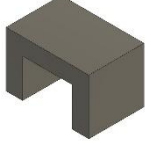
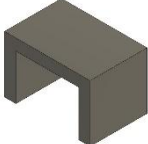


3.1 SELEÇÃO E PREPARAÇÃO DAS GEOMETRIAS

O treinamento da rede neural foi executado usando um conjunto de geometrias em seus estados iniciais e seus respectivos estados deformados. Este par formado pelas geometrias iniciais e deformadas é referido neste trabalho como exemplo de treinamento, visto que são utilizados para ensinar à rede neural o comportamento dimensional de peças quando fundidas em areia. Essas geometrias foram elaboradas pelo autor, com formas e dimensões propícias para a ocorrência das principais deformações que ocorrem na fundição, fornecendo à rede neural dados de treinamento que expressem de forma eficiente o comportamento que esta deverá aprender.

3.1.1 Projeto das geometrias iniciais

As geometrias foram baseadas em outros trabalhos científicos que propunham geometrias para estudar e apresentar o comportamento dimensional de peças fundidas. A Figura 17 mostra as geometrias no estado inicial usadas como exemplos de treinamento, bem como a respectiva fonte da geometria.

Figura 17 – Geometrias usadas para treinamento da rede neural

			
Geometria 1 Fonte: Campbell (2011)	Geometria 2 Fonte: Campbell (2011)	Geometria 3 Fonte: Campbell (2011)	Geometria 4 Fonte: Campbell (2011)
			
Geometria 5 Fonte: Campbell (2011)	Geometria 6 Fonte: Campbell (2011)	Geometria 7 Fonte: Campbell (2011)	Geometria 8 Fonte: Campbell (2011)
			
Geometria 9 Fonte: Campbell (2011)	Geometria 10 Fonte: o autor	Geometria 11 Fonte: o autor	Geometria 12 Fonte: o autor
			
Geometria 13 Fonte: o autor	Geometria 14 Fonte: o autor	Geometria 15 Fonte: Galles, 2017	Geometria 16 Fonte: Galles, 2017
			
Geometria 17 Fonte: Galles, 2017	Geometria 18 Fonte: Galles, 2018	Geometria 19 Fonte: Thiel e Ravi, 2016	

Fonte: Elaborada pelo Autor (2025).

Todas as geometrias foram selecionadas pela sua suscetibilidade a algum comportamento dimensional durante a fundição em areia. As geometrias de número 1 a 4, por exemplo, são suscetíveis principalmente ao comportamento da restrição da areia na redução da contração linear, abordada na seção 2.1.1. Já as geometrias 15 a 17 são algumas das geometrias que são suscetíveis a empenamentos causados pela restrição da areia, tema também abordado na seção 2.1.1. As geometrias 6 a 9, segundo o respectivo autor, apresentam distorções relacionadas o resfriamento não uniforme durante o resfriamento, comportamento explicado na seção 2.1.3. E as geometrias 18 e 19 foram escolhidas por apresentarem deformações devido a dilatação da areia, fenômeno explicado na seção 2.1.2. Além disso, todas as geometrias, da 1 a 19, estão sujeitas a contração térmica. Dessa forma, cada uma das 19 geometrias representa ao menos um dos principais comportamentos de deformação em fundição.

3.1.2 Obtenção das geometrias deformadas

As geometrias no estado deformado foram obtidas com a simulação de fundição em areia no software Magmasoft. Nesta etapa, foram configurados os materiais das peças e dos moldes, bem como parâmetros do processo de fundição considerados na simulação.

O material usado na simulação foi o ferro fundido cinzento GJL-250 da biblioteca de materiais do software, cuja composição química é mostrada na Tabela 1 e as curvas das principais propriedades mecânicas e físicas são mostradas no Apêndice A. Este ferro fundido cinzento foi utilizado neste estudo devido a apresentar maior deformação nas peças fundidas em relação a outras ligas comuns de ferros fundidos, como o nodular e o vermicular.

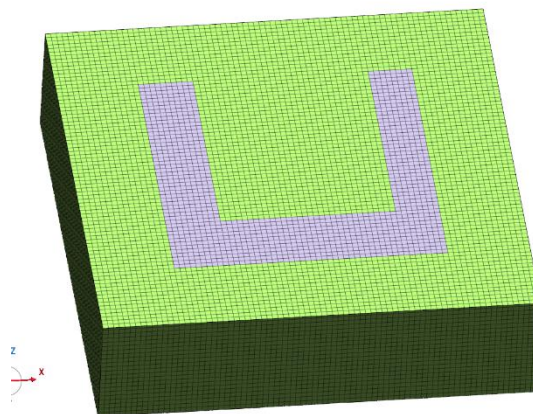
Tabela 1 – Composição química do material utilizado

C	Si	Mn	P	S	Cr	Cu
3,25%	2,05%	0,7%	0,15%	0,1%	0	0
Sn	Mg	Mo	Ni	Ce	Sb	N
0	0	0	0	0	0	0

Fonte: Elaborado pelo Autor (2025)

As geometrias foram carregadas no programa no formato de arquivo STL, e o centro do sistema de coordenadas da peça foi mantido coincidente com o do programa. Em seguida o molde foi adicionado com formato cúbico ultrapassando a dimensão das peças em 30mm em todas as direções, como pode ser visto na Figura 18. Visto que as geometrias possuem dimensões externas variadas, o molde foi ajustado para cada uma das 19 geometrias a fim de manter a espessura de parede igual entre as geometrias. O molde então teve o volume da peça subtraído de si automaticamente pelo algoritmo do programa, que utiliza a sequência dos elementos para realizar operações booleanas. Nenhum canal de alimentação, massalote, ou qualquer outro acessório metalúrgico foi utilizado na simulação.

Figura 18 – Configuração para simulação de uma das geometrias



Fonte: Elaborada pelo Autor (2025).

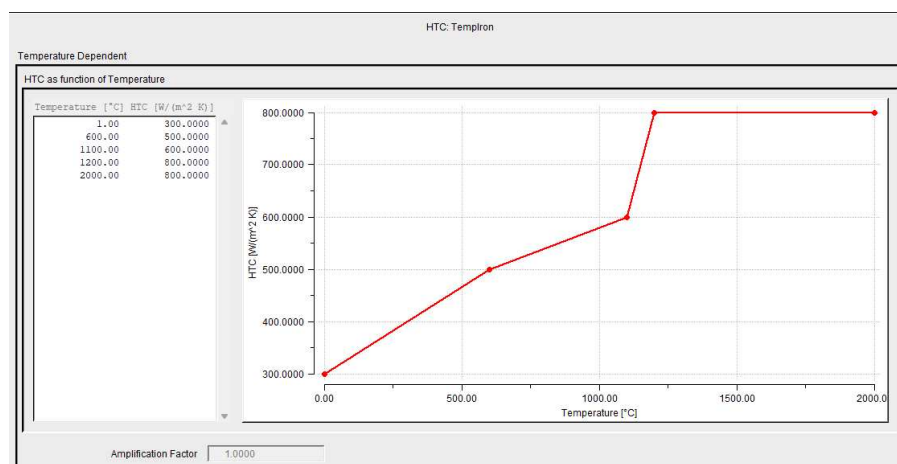
A discretização do volume de simulação, mostrada na Figura 18, foi feita com uma malha de elementos cúbicos de 2mm de lado, tanto para a peça quanto para o molde, resultando em um número de elementos que variou entre as geometrias desde 62,5 mil até 648 mil elementos. Já a discretização do tempo é feita de forma automática pelo programa, não sendo configurável pelo usuário. O mecanismo de término da simulação foi o atingimento da temperatura de 100°C no ponto mais quente da peça.

A primeira condição de contorno utilizada na simulação foi a temperatura do metal configurada em 1400°C para todos os elementos de malha pertencentes a peça. A temperatura inicial do molde foi configurada em 40°C,

também de forma homogênea em todos os elementos da malha pertencentes ao molde. A velocidade inicial do metal foi configurada igual a zero, e nenhuma diferença de pressão nem geração de metal foi adicionada a fim de simular apenas o comportamento do resfriamento das peças com o metal partindo de um estado líquido-superaquecido.

Na interface entre o metal e o molde, devido ao contato, foi utilizado uma condutância térmica de contato dependente da temperatura, cuja curva é mostrada na Figura 19. Para este dado foi utilizada a configuração *Templron* da biblioteca de materiais do programa. Já na interface entre o molde e o ambiente externo, foi utilizada a configuração *default_environment*, a qual é bloqueada pelo programa.

Figura 19 – Curva da condutância térmica de contato em função da temperatura



Fonte: Elaborada pelo Autor (2025).

A simulação de tensões foi executada juntamente com a simulação de resfriamento, não sendo necessárias configurações diferentes para a malha ou parâmetros dos materiais, somente a indicação de quais corpos devem ser incluídos na simulação e se são deformáveis ou não. Neste caso, tanto a peça quanto o molde foram incluídos na simulação e ambos considerados deformáveis.

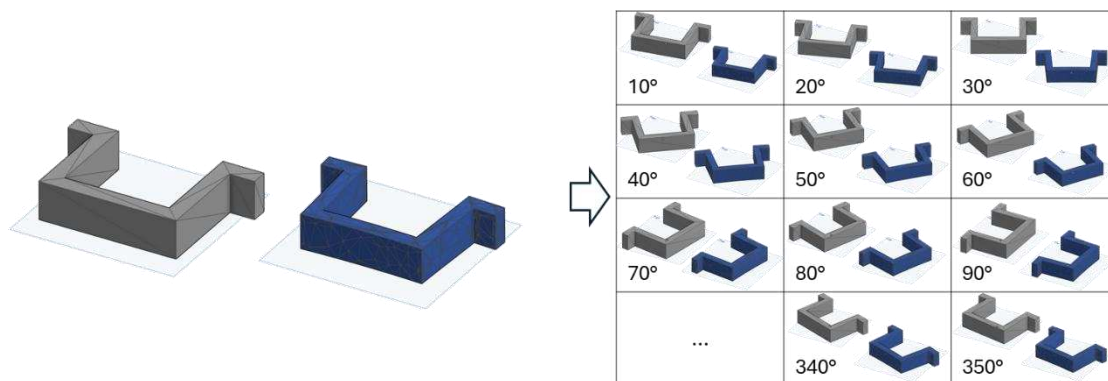
Ao final da simulação, com os resultados da simulação de tensões, a geometria deformada foi apresentada pelo software considerando os deslocamentos ocorridos durante o resfriamento. Estes deslocamentos foram

ampliados em 10 vezes para gerar a geometria no estado deformado utilizada no treinamento da rede neural.

3.1.3 Programa para aumento dos dados de treinamento

O aprendizado da rede neural é prejudicado se houverem poucos exemplos de treinamento, portanto, a fim de aumentar o número de amostras, as geometrias foram rotacionadas em torno do eixo Z em instâncias de 10 graus, dessa forma obtendo 36 vezes mais geometrias, totalizando 684 pares de geometrias, ou exemplos de treinamento.

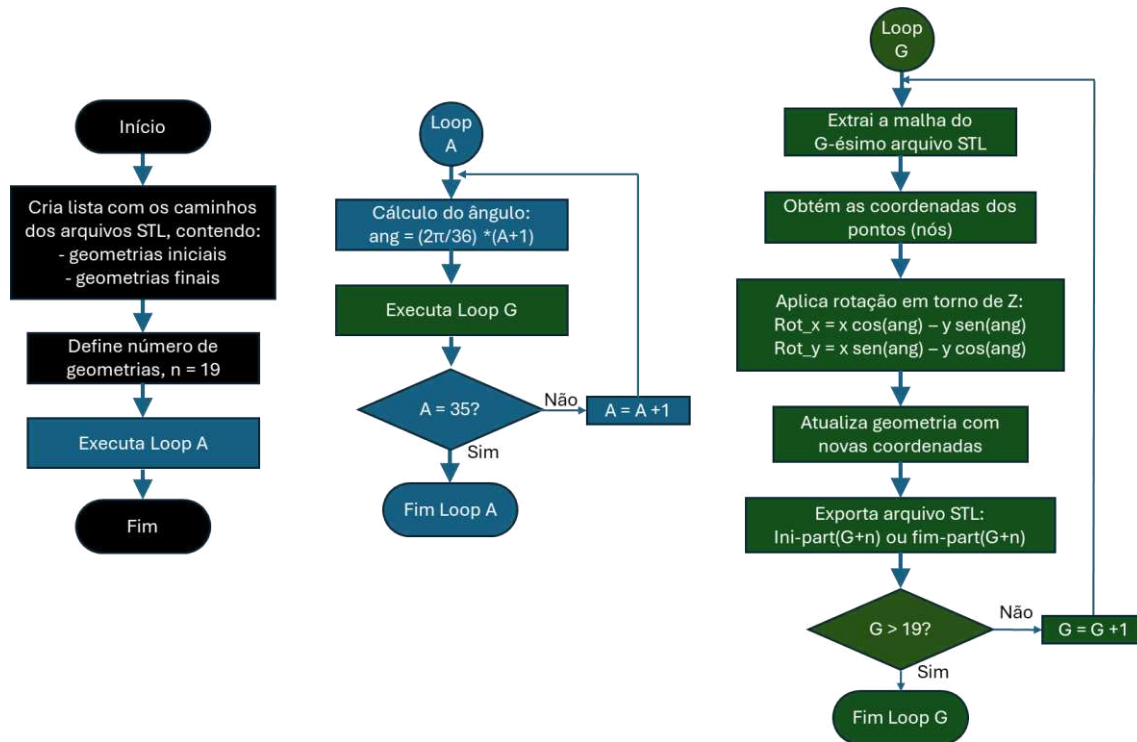
Figura 20 – Aumento de dados por rotação em torno do eixo Z



Fonte: Elaborada pelo Autor (2025).

As réplicas foram criadas por meio de um programa em *Python* cujo algoritmo é mostrado no fluxograma da Figura 21. O programa inicia criando uma lista com os caminhos dos arquivos STL que contém as geometrias iniciais e deformadas. Em seguida o *loop* A, que percorre os ângulos de rotação, é executado 35 vezes a fim de criar novas geometrias espaçadas em 10° entre si, isto é, as geometrias originais em 0°, a primeira repetição em 10°, segunda repetição em 20°, até a repetição número 35, que rotaciona as geometrias em 350°.

Figura 21 – Fluxograma do programa de multiplicação das geometrias para aumento de dados



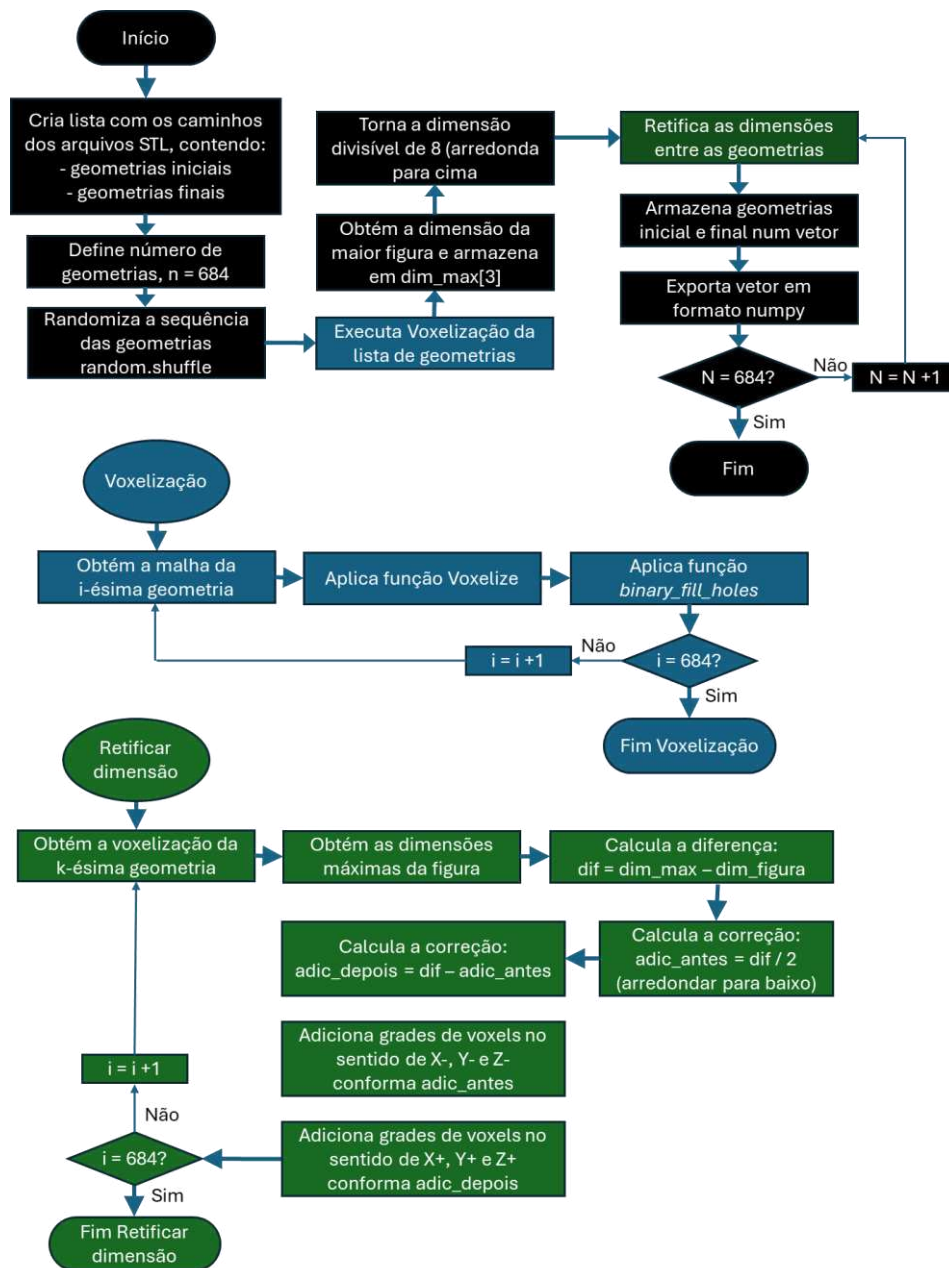
Fonte: Elaborada pelo Autor (2025).

Dentro do *loop A*, o *loop G*, que é responsável por percorrer todas as 19 geometrias, é executado uma vez para cada geometria, rotacionando-as e exportando os novos arquivos STL. Assim, todas as geometrias são rotacionadas em 10° e depois todas são rotacionadas em 20° e assim por diante até a última instancia. Dessa forma, o conjunto de dados criado não contém repetições da mesma geometria em sequência, a fim de prevenir alguma geração de tendência durante o treinamento da rede neural.

3.1.4 Programa de pré-processamento dos dados

Após a multiplicação das geometrias para aumento de dados, estas foram pré-processadas num outro programa em *Python*, cujo fluxograma é mostrado na Figura 22.

Figura 22 – Fluxograma do programa de pré-processamento dos dados



Fonte: Elaborada pelo Autor (2025).

O pré-processamento inicia com a criação da lista de arquivos contendo as geometrias de treinamento, que nesta etapa é de 684 geometrias iniciais e o mesmo número de geometrias deformadas. Após a criação da lista, 5% das geometrias são reservadas para formar o conjunto de teste do programa, conforme será citado na seção 3.1.5. Estas geometrias selecionadas foram as geometrias de número 1 a 19 e todas as geometrias cujo número fosse múltiplo

de 44, até o limite de 684. Dessa forma, as geometrias reservadas para o conjunto de testes foram as mostradas na Tabela 2.

Tabela 2 – Geometrias separadas para teste

Nº geometria	Cópia da geometria:	Rotação	Nº geometria	Cópia da geometria:	Rotação
1	-	0°	44	6	20°
2	-	0°	88	12	40°
3	-	0°	132	18	60°
4	-	0°	176	5	90°
5	-	0°	220	11	110°
6	-	0°	264	17	130°
7	-	0°	308	4	160°
8	-	0°	352	10	180°
9	-	0°	396	16	200°
10	-	0°	440	3	230°
11	-	0°	484	9	250°
12	-	0°	528	15	270°
13	-	0°	572	2	300°
14	-	0°	616	8	320°
15	-	0°	660	14	340°
16	-	0°			
17	-	0°			
18	-	0°			
19	-	0°			

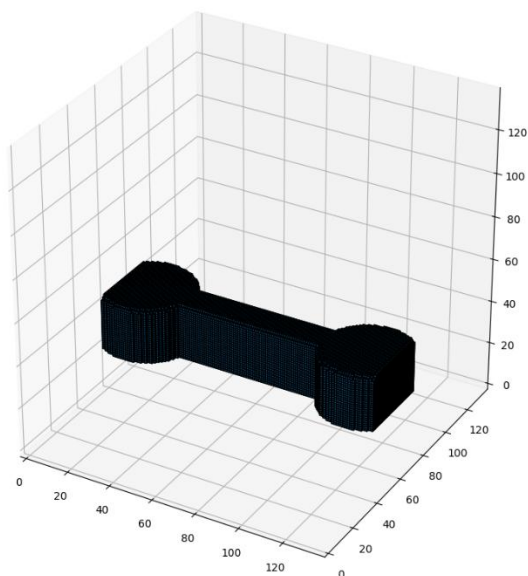
Fonte: Elaborado pelo Autor (2025).

Dessa forma, são selecionados para o conjunto de testes e separadas as geometrias de 1 a 19, e aquelas cujo número era divisível por 44, a fim de selecionar todas as geometrias originárias e algumas rotacionadas razoavelmente distribuídas pelo restante do conjunto restante.

A lista das geometrias de treinamento e validação foi então randomizada, a fim de desfazer qualquer sequência que possa gerar uma tendência durante o treinamento. Com a lista randomizada, o programa segue com a execução da função de voxelização, que busca os arquivos STL no diretório, obtém a malha de cada um deles e a voxeliza usando a ferramenta *voxelize*, da biblioteca *Trimesh*. Esta função, porém, ativa apenas os voxels da periferia da peça, criando uma geometria com o interior vazio, semelhante a uma casca.

Para transformar a casca gerada pela função *voxelize*, foi usada a função *binary_fill_holes*, da biblioteca *SciPy*. Este comando identifica os contornos fechados e preenche-os com voxels ativos, tornando a geometria maciça.

Figura 23 – Voxelização da geometria número 3



Fonte: Elaborada pelo Autor (2025).

A rede neural, porém, requer que os exemplos de treinamento tenham as mesmas dimensões externas, isto é, o número de voxels nas direções X, Y e Z sejam iguais entre os exemplos de treinamento. Para ajustar os dados de entrada foi utilizada a função denominada *Retificar_dimensao*, a qual obtém as geometrias voxelizadas e calcula e aplica a correção na dimensão necessária para que as amostras de treinamento tenham todas a mesma dimensão e que esta dimensão seja divisível por 16, visto que posteriormente, na rede neural, serão utilizadas funções de redução e ampliação dimensional. Para a aplicação das correções, a ferramenta *Pad*, da biblioteca *NumPy*, foi utilizada para adicionar voxels nulos ao redor da geometria, mantendo-a centralizada no volume de controle.

Esses tensores tridimensionais têm um tamanho considerável, o que torna o programa de treinamento da rede neural suscetível a falha por excesso de uso da memória RAM da GPU se o programa mantiver na memória todas as geometrias de treinamento ao mesmo tempo. Para contornar este problema,

todos os tensores contendo as geometrias pré-processadas gerados por este programa são exportados no formato numpy (.npy), pois desta forma esses tensores podem posteriormente ser carregados no programa de treinamento da rede neural à medida que a rede demanda e somente um por vez, isto é, em regime de *streaming*. Dessa forma, apenas um exemplo de treinamento é mantido na memória do programa, evitando este problema de uso excessivo de memória RAM. Os arquivos npy foram carregados no Google Drive, de onde o Google Colab pode acessar diretamente durante o treinamento da rede neural.

3.1.5 Divisão dos dados

As 684 geometrias foram divididas em três conjuntos para possibilitar o treinamento da rede neural, sua validação e teste. 34 geometrias (equivalente a 5%) são reservadas para formarem o conjunto de teste, mantidas separadas para avaliar a capacidade de generalização do modelo em dados não vistos. Das outras 650 geometrias, o conjunto de treinamento ocupa 85%, sendo 552 no total, utilizadas para ajustar os pesos e parâmetros do modelo. Outros 15% das 650 geometrias, isto é, 98 delas, compuseram o conjunto de validação, permitindo monitorar o desempenho durante o treinamento e detectar sobreajuste, caso houvesse.

3.2 PROGRAMA DE TREINAMENTO DA REDE NEURAL

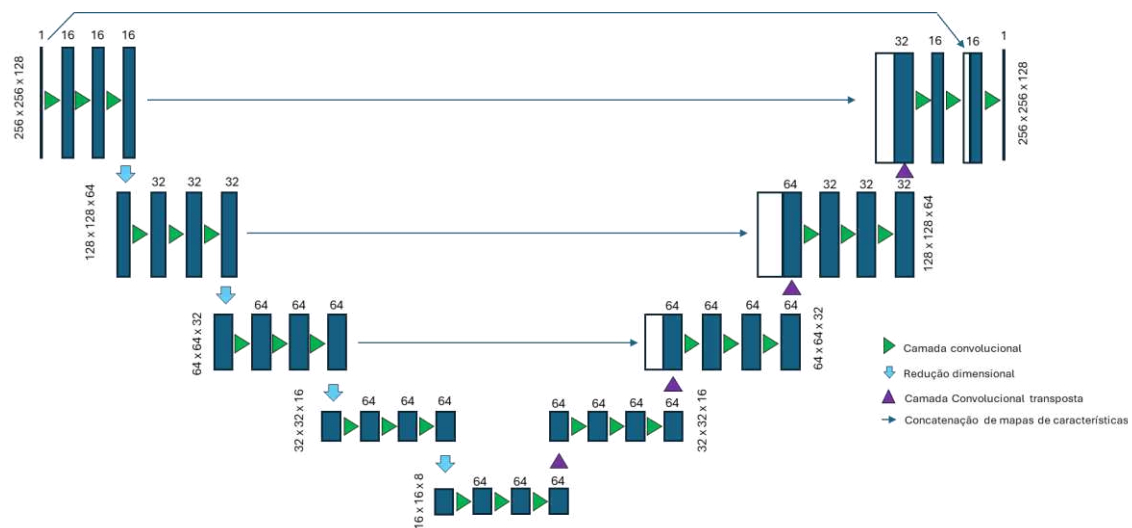
Foi elaborado um programa para executar o treinamento da rede neural, projetado para integrar todas as etapas essenciais ao processo de aprendizado supervisionado aplicado à previsão de distorções em geometrias tridimensionais. Neste programa a arquitetura da rede neural é definida juntamente com os outros parâmetros de treinamento, e este é responsável por treinar a rede neural e exportá-la após o treinamento para possibilitar o uso posterior. O fluxograma deste programa é mostrado no Apêndice C.

3.2.1 Arquitetura da rede neural

Nesta seção é apresentada a arquitetura da rede neural desenvolvida para prever as deformações nas geometrias, utilizando uma adaptação da arquitetura U-Net com convoluções 3D. A rede é organizada em dois trechos

principais: codificação (*encoder*), responsável pela extração de características das geometrias, e decodificação (*decoder*), voltada para a reconstrução e previsão das deformações.

Figura 24 – Arquitetura da rede neural



Fonte: Elaborada pelo Autor (2025).

Todas as camadas convolucionais empregam filtros de tamanho $3 \times 3 \times 3$, inicialização HeNormal, preenchimento com *padding* do tipo "same" e normalização por lote (Batch Normalization). Na parte de codificação, a ativação utilizada é a função *Leaky ReLU* (com inclinação negativa de 0,1), enquanto que na parte de decodificação é utilizada a função *ReLU*.

A codificação é composta por quatro níveis sequenciais, cada um contendo três camadas convolucionais, aumentando gradualmente a profundidade das características extraídas (16 filtros no primeiro nível, 32 no segundo, 48 nos terceiro e quarto níveis). Após cada nível, aplica-se uma operação de subamostragem por média (*MaxPooling3D*) com janela $2 \times 2 \times 2$ e *dropout* de 30% para combater o sobreajuste.

No quinto nível, o mais profundo da rede, são realizadas três camadas convolucionais adicionais com 48 filtros para captar características mais complexas, também seguidas por *dropout* de 30%.

A decodificação utiliza camadas convolucionais transpostas (*Conv3DTranspose*) com *strides* de $2 \times 2 \times 2$ para expandir a dimensão da imagem e restaurar

progressivamente as dimensões espaciais originais das geometrias. Em cada nível, os mapas de características resultantes são concatenados com os mapas correspondentes do *encoder* ao longo do eixo dos canais, a fim de melhorar recuperação das informações espaciais e facilitando a propagação dos gradientes.

As camadas de decodificação reduzem o número de filtros de forma gradual (64, 32 e posteriormente 16), até que as dimensões originais sejam recuperadas. Antes da camada de saída, ocorre uma última concatenação com a entrada original para aprimorar a capacidade da rede neural em obter os detalhes finos das imagens.

Por fim, a camada de saída é uma convolução 3D com um único filtro, inicialização *GlorotNormal* e ativação *sigmoid*. A saída gerada representa um mapa volumétrico onde cada voxel contém a probabilidade de pertencer ou não ao material previsto após deformação, que é posteriormente binarizada para formar a geometria final prevista pela rede neural. A binarização é feita usando um limiar de 0,5, ou seja, valores abaixo desse limiar são considerados como 0 (sem material naquele ponto) e valores acima são considerados como 1 (com material naquele ponto).

3.2.2 Otimização e Função de Erro

Para o treinamento da rede neural, optou-se pela utilização do otimizador Adam com uma taxa de aprendizado inicial de 0,0001. Um parâmetro de limitação de norma dos gradientes (*clipnorm*) foi definido como 1,0 para prevenir o aumento descontrolado dos valores dos pesos, e prover mais estabilidade ao treinamento da rede neural.

Foi implementada também uma estratégia de ajuste dinâmico da taxa de aprendizado por meio de uma função de redução adaptativa (*Reduce_LR*), que reduzia a taxa de aprendizado à metade sempre que o valor de erro calculado não apresentava melhoria após três épocas consecutivas de treinamento. Essa técnica, conhecida como redução adaptativa da taxa de aprendizado (*adaptive learning rate scheduling*) foi usada para que o modelo pudesse realizar ajustes

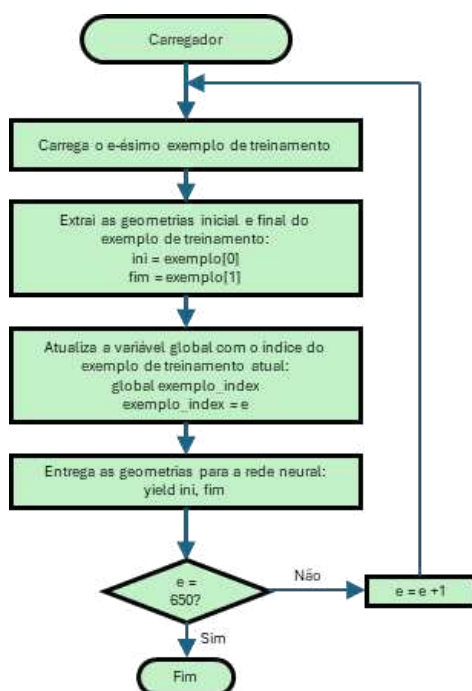
mais finos nos pesos conforme se aproxima de um ponto de mínimo da curva de erro.

Para o cálculo do gradiente, a função de erro utilizada foi a Entropia Cruzada Binária (*Binary Crossentropy* - BCE), fornecida pela biblioteca Keras. Essa função é amplamente empregada em tarefas de classificação binária, pois permite calcular de forma eficiente o erro da previsão ao penalizar, em escala logarítmica, a diferença entre o valor predito e o valor esperado em cada voxel.

3.2.3 Carregamento dos dados sob demanda

À medida que o treinamento da rede neural é executado, os exemplos de treinamento são importados ao programa a fim de otimizar memória RAM, conforme já comentado na seção 3.2.5. Estes exemplos de treinamento são importados do Drive através da função de carregamento dos dados denominada Carregador. Esta função, cujo fluxograma é mostrado na Figura 25, é utilizado no programa de treinamento da rede neural e é acionada pelo *loop* de treinamento à medida que este processa os exemplos de treinamento.

Figura 25 – Fluxograma da função de carregamento dos dados



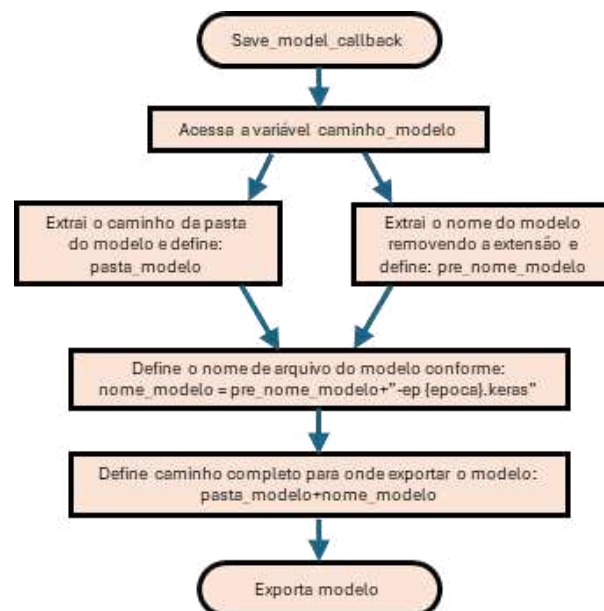
Fonte: Elaborada pelo Autor (2025).

Dada a demanda por uma nova geometria, a função Carregador busca e importa o arquivo contendo o exemplo de treinamento pré-processado e extrai as geometrias inicial e final em seguida. O índice correspondente a geometria é atualizado numa variável global para possibilitar ao programa saber qual geometria está sendo disponibilizada. Após entregar as geometrias para o *loop* de treinamento, o processo é repetido até que se esgotem os 650 exemplos de treinamento e validação.

3.2.4 Função para exportar rede neural

Durante o treinamento, a rede neural é modificada à medida que os pesos são recalculados, como abordado na seção 2.3.3. Para possibilitar a obtenção de dados mais completos na etapa de testes, a rede neural foi exportada após finalizar cada época do treinamento. Para isso, foi utilizada uma função para exportação da rede neural, denominada neste contexto *save_model_callback*.

Figura 26 – Fluxograma da função de exportação da rede neural



Fonte: Elaborada pelo Autor (2025).

O programa inicia identificando o caminho completo para onde o modelo deve ser exportado, neste caso direcionado para uma pasta no Google Drive.

Em seguida o programa separa o caminho completo no caminho para a pasta para onde o modelo deve ser exportado e no prenome do arquivo. O prenome é definido como o nome geral do modelo, definido pelo usuário, neste caso com “Modelo”. Este prenome é usado para definir o nome de arquivo de arquivo do modelo em cada época, compondo o prenome com o sufixo “-ep{epoca}.keras”, onde “{epoca}” é a variável que contém o número da época em treinamento. Após isso, o caminho completo do arquivo é formado compondo o caminho completo para a pasta com nome de arquivo do modelo.

3.2.5 Recursos computacionais

Todo o desenvolvimento e execução dos programas de pré-processamento, treinamento e teste da rede neural foram realizados no ambiente de programação Google Colab. Para o treinamento da rede neural foi utilizada a GPU remota NVIDIA A100 e para os programas de teste a GPU remota NVIDIA T4 foi utilizada. Já para os programas de pré-processamento, não foi utilizada GPU remota, apenas o processamento da CPU.

A GPU A100 dispõe de 40 GB de memória RAM dedicada, o que permitiu o processamento eficiente do treinamento da rede neural bem como a alocação das geometrias utilizadas pelo programa durante a execução. A GPU T4 dispõe de 15GB de memória RAM, e ofereceu uma velocidade elevada de processamento quando comparada a CPU.

3.3 PÓS-PROCESSAMENTO E MEDIÇÃO DA PRECISÃO DIMENSIONAL

Após o treinamento do modelo, tornou-se necessário avaliar seu desempenho utilizando os dados reservados para o conjunto de teste.

3.3.1 Avaliação do conjunto de testes

O processo de avaliação foi automatizado por meio de um programa desenvolvido em Python, cujo fluxograma é apresentado no Apêndice E, que percorre as diferentes épocas de treinamento da rede neural e realiza predições sobre as geometrias do conjunto de teste.

O programa inicia carregando os caminhos de entrada, gerando a lista com os números das geometrias reservadas para teste e organizando a sequência de épocas que serão avaliadas. Em seguida, realiza o pré-processamento das geometrias por meio da voxelização dos arquivos STL e correção dimensional, assegurando que todas as entradas estejam compatíveis com a arquitetura da rede neural. Para cada época especificada, o modelo correspondente é carregado e utilizado para realizar previsões sobre as geometrias de teste. As saídas previstas são binarizadas com limiar de 0,5, convertidas em malhas de superfície por meio do algoritmo *marching cubes* e exportadas no formato STL. Em paralelo, são calculadas métricas quantitativas de desempenho como F1-score e erro (BCE). O programa também contempla verificações de integridade das previsões, controle de erros durante o carregamento dos modelos e a tecelagem das geometrias.

Para cada geometria e para cada época, a previsão é comparada com sua respectiva geometria deformada, sendo o F1-score calculado pela função específica. O programa registra os valores obtidos e armazena os dados em arquivos de saída para posterior análise. O objetivo dessa etapa é determinar o desempenho da rede neural ao longo do processo de treinamento de forma sistemática e padronizada.

3.3.1.1 Seleção da Melhor Época de Treinamento

Após a coleta dos F1-scores de todas as épocas avaliadas, foi realizada a seleção da época com melhor desempenho, com base no maior valor médio de F1-score obtido sobre o conjunto de teste. A rede neural correspondente a essa época foi, então, adotada como modelo final para as etapas subsequentes da metodologia.

Em seguida, o valor de F1-score de cada geometria foi avaliado e usado para selecionar um conjunto de geometrias para uma avaliação dimensional detalhada utilizando o software CloudCompare, conforme descrito na próxima seção.

3.3.2 Avaliação Dimensional com CloudCompare

Além da análise quantitativa baseada em métricas globais das geometrias, foi conduzida uma avaliação dimensional local com o objetivo de analisar a qualidade espacial das predições feitas pela rede neural. Essa etapa foi realizada com o software CloudCompare, especializado em comparação tridimensional entre nuvens de pontos e malhas trianguladas.

Para cada caso analisado, foram utilizadas duas versões da geometria: a geometria predita, gerada pela rede neural selecionada pelo procedimento descrito na seção anterior, e a geometria deformada de referência, proveniente da simulação convencional, mas submetida previamente à voxelização e tesselação, de modo a isolar os erros atribuíveis ao processo neural, excluindo os efeitos relacionados à voxelização e mantendo somente a performance da rede neural.

Ambas as geometrias foram importadas no *CloudCompare*, onde foi realizado o alinhamento automático entre elas por meio do algoritmo *Iterative Closest Point* (ICP), que minimiza iterativamente a distância quadrática entre superfícies até atingir convergência. O critério de parada adotado foi a estabilização do valor de erro RMS (*Root Mean Square*), com incremento inferior a 10^{-6} .

Uma vez alinhadas, foi aplicada a ferramenta de comparação ponto-a-malha (*Cloud-to-Mesh*, abreviado pelo software como C2M). Essa operação calcula, para cada ponto da nuvem da geometria predita, a menor distância até a superfície da malha da geometria de referência. O valor obtido representa então o desvio dimensional local, indicando se houve excesso ou falta de material na previsão da forma final.

Esses desvios foram representados por meio de um mapa de cores aplicado sobre a superfície da peça, adotando o seguinte esquema cromático:

- Desvios próximos de zero foram representados em verde;
- Para valores positivos (excesso de material), foi adotado um gradiente do verde até amarelo em +1 mm, e vermelho para valores acima de +2 mm;
- Para valores negativos (falta de material), o gradiente foi do verde até azul para desvios de -1 mm ou menos.

Adicionalmente, foram inseridos pontos de medição sobre a superfície das geometrias, em regiões tecnicamente relevantes, indicando os valores locais de desvio dimensional em milímetros. Isso permitiu destacar áreas críticas e facilitar a leitura direta dos erros pontuais.

3.3.2.1 Análise Estatística por Histograma de Desvios

Complementando a visualização espacial, foi gerado um histograma de distribuição dos desvios dimensionais para cada geometria analisada. No gráfico, o eixo horizontal representa os valores de desvio dimensional (em milímetros), enquanto o eixo vertical indica o número de pontos da nuvem que apresentam aquele valor específico de desvio. Dessa forma, é possível observar se os erros se concentram próximos de zero (bom desempenho) ou se estão distribuídos em faixas amplas, indicando falhas localizadas ou sistemáticas.

3.3.2.2 Análise de sensibilidade ao refinamento da voxelização

Com o intuito de avaliar a sensibilidade da rede neural à discretização espacial das geometrias de entrada, foi conduzido um experimento sistemático em que se variou o tamanho dos voxels utilizados no conjunto de treinamento. Essa análise, inspirada nos princípios dos testes de sensibilidade de malha aplicados em simulações numéricas, visou investigar o impacto do grau de refinamento na qualidade das predições fornecidas pela rede. Três níveis de discretização foram adotados: 0,7 mm, 0,8 mm e 0,9 mm, mantendo-se constantes todas as demais variáveis do processo de treinamento, como arquitetura da rede, hiperparâmetros e número de épocas

Para cada nível de refinamento, a rede neural foi treinada do zero com os respectivos dados voxelizados, gerando modelos distintos. Em seguida, os três modelos foram testados sobre o mesmo conjunto fixo de geometrias, composto por 34 peças reservadas para teste. O desempenho de cada modelo foi avaliado por meio do F1-score, calculado individualmente para cada geometria.

4 RESULTADOS

O Capítulo 4 apresenta os resultados obtidos ao longo do desenvolvimento do trabalho. Inicialmente, são analisadas as simulações realizadas no software MAGMASOFT, com foco na reprodutibilidade dos efeitos de diferentes fenômenos físicos sobre as geometrias. Em seguida, são discutidos os procedimentos de pré-processamento aplicados às geometrias utilizadas na rede neural. Posteriormente, são apresentados os resultados do treinamento da rede, com análise do erro e do F1-score. Por fim, avalia-se o desempenho da rede no conjunto de testes, incluindo critérios para seleção da melhor época de treinamento e a análise das geometrias tesselladas.

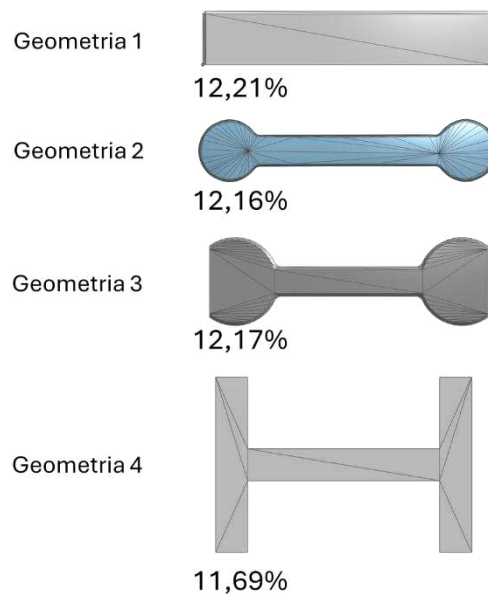
4.1 ANÁLISE DOS RESULTADOS DA SIMULAÇÃO DO MAGMASOFT

A simulação realizada no software Magmasoft teve como objetivo reproduzir os comportamentos físicos esperados durante a solidificação de geometrias fundidas, como a contração térmica e o efeito da restrição da areia. Nas subseções a seguir serão relatadas as conformidades e não conformidades dos resultados obtidos via simulação convencional com o descrito pela literatura.

4.1.1 Análise da reprodutibilidade do efeito da restrição da areia na contração das geometrias

Campbel (2011), conforme descrito na seção 2.1.1, usa geometrias semelhantes às geometrias 1, 2, 3 e 4, mostradas na Figura 27 para mostrar a diferença na contração de peças sujeitas a diferentes graus de resistência da areia. A geometria 1 está sujeita a nenhuma restrição de areia, e, portanto, deve apresentar a maior redução no comprimento enquanto a geometria nº 4, pelo cálculo proposto pelo autor, deveria apresentar uma contração menor que a metade da primeira.

Figura 27 - Comparação da redução de comprimento das geometrias 1, 2, 3 e 4



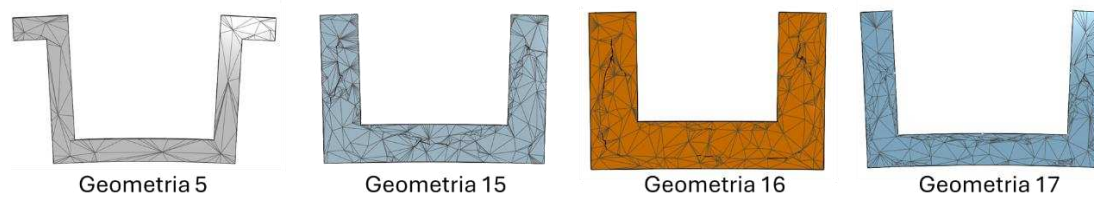
Fonte: Elaborada pelo Autor (2025).

Portanto, o efeito da restrição da areia na contração das peças estava incluído nas geometrias deformadas, porém não ocorreu da forma descrita na literatura por Campbell (2011).

4.1.2 Análise da reprodutibilidade do efeito da restrição da areia na distorção das geometrias

A geometria 5, 15, 16 e 17 foram projetadas para representar um cenário em que o efeito de restrição da areia do molde tende a causar distorção das peças, conforme descrito por Campbell (2011) e Galles e Beckerman (2017). Nas simulações realizadas com o Magmasoft, cujos resultados são mostrados na Figura 28, as geometrias 5 e 17 exibiram uma distorção acentuada das hastes laterais, enquanto as geometrias 15 e 16 mostraram uma distorção menor, o que são comportamentos alinhados com o esperado.

Figura 28 - Geometria n° 5, 15, 16 e 17 nos estados deformados calculados pelo Magmasoft



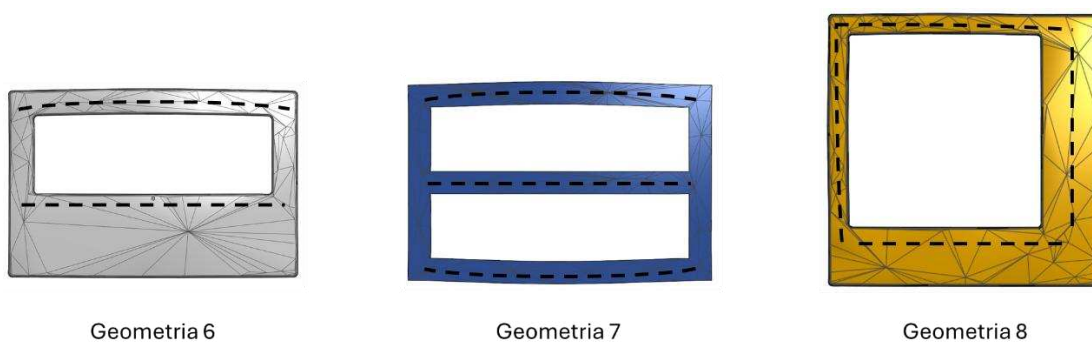
Fonte: Elaborada pelo Autor (2025).

Dessa forma, as geometrias descritas compuseram exemplos de treinamento que continham o comportamento da distorção de peças devido à restrição da areia.

4.1.3 Análise da reprodutibilidade do efeito do resfriamento não uniforme

As geometrias 6, 7 e 8 mostradas na Figura 29 foram desenvolvidas com base nos princípios de fundição de seções mistas abordados por Campbell (2011). O autor destaca que peças com variações abruptas de espessura estão sujeitas ao resfriamento desigual entre as seções finas e espessas, o que pode resultar em grandes deformações ou rupturas. (Campbell, 2011)

Figura 29 - Estados deformados das geometrias 6, 7 e 8



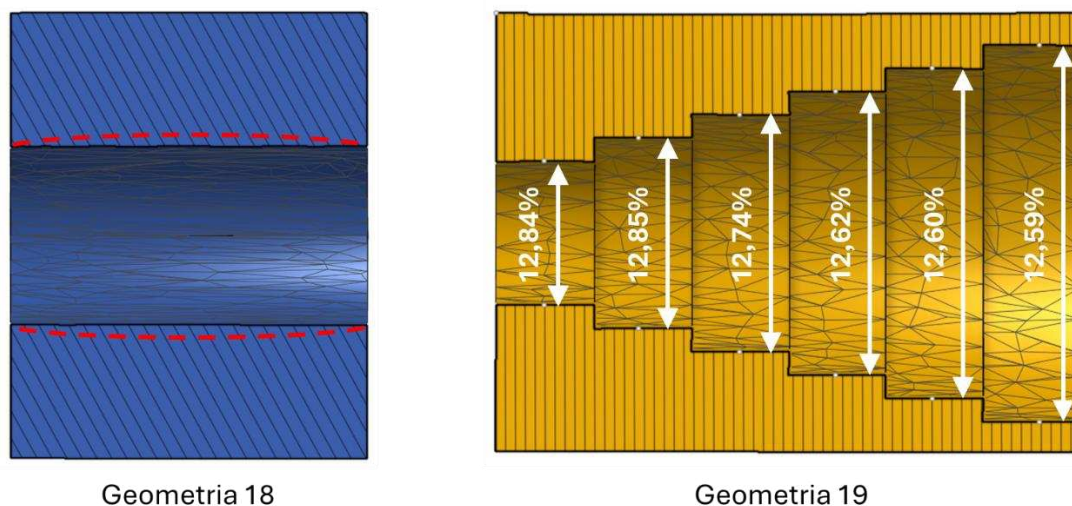
Fonte: Elaborada pelo Autor (2025).

Os resultados das simulações para essas geometrias corroboraram o comportamento esperado, mostrando deformações maiores nas regiões que tendem a solidificar antes das outras.

4.1.4 Análise da reprodutibilidade do efeito da dilatação do molde

A geometria número 18, mostrada na Figura 30 foi baseada no trabalho de Galles e Beckermann (2015) com o objetivo de reproduzir o efeito da dilatação da areia do molde, que deveria resultar em uma superfície interna com formato de barril — apresentando um diâmetro maior na região central da peça e diâmetros menores nas extremidades. No entanto, esse comportamento não foi observado nos resultados da simulação. (Galles; Beckermann, 2015)

Figura 30 - Geometrias 18 e 19 nos estados deformados



Fonte: Elaborada pelo Autor (2025).

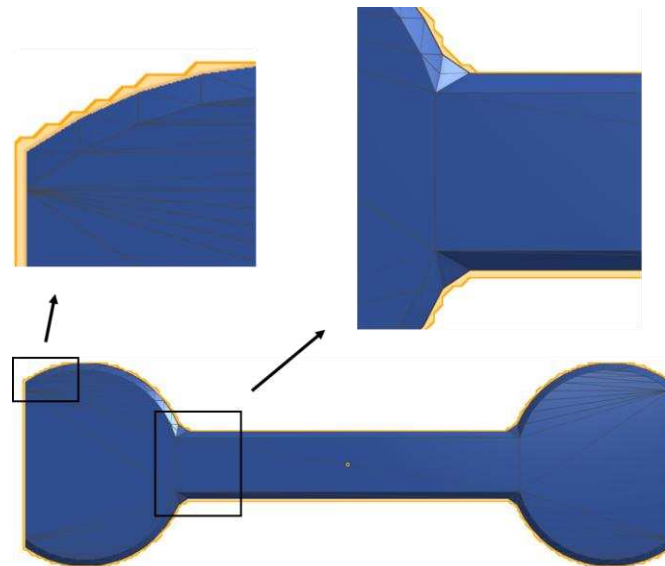
De forma semelhante, na geometria 19, segundo Thiel e Ravi (2016), esperava-se uma variação dimensional menor do que a apresentada nos diâmetros maiores, causada pelo maior aquecimento da areia nas regiões de diâmetros menores, acentuando o fenômeno de dilatação da areia. Entretanto, esse comportamento também não foi reproduzido na simulação. (Thiel; Ravi, 2016)

4.2 ANÁLISE DO PRÉ-PROCESSAMENTO DAS GEOMETRIAS

Durante o processo de voxelização das geometrias, foi observado um problema de aumento não intencional nas dimensões das peças após a

conversão para o formato voxelizado. Na geometria número 4, ao ser voxelizada e tessellada novamente para avaliação, foi possível observar perda da curvatura e aparecimento de serrilhado e faces planas em regiões que eram redondas, como mostrado na Figura 31.

Figura 31 - Análise da voxelização da geometria número 4



Fonte: Elaborada pelo Autor (2025).

Além disso a peça teve seu comprimento aumentado de 190 para 191,2mm, espessura aumentada de 35 para 36mm e a largura se manteve constante em 60mm. O deslocamento da região central da peça que pode ser percebido na Figura 31, é de 0,8mm caracterizado pelo deslocamento de somente uma das paredes da peça. Comportamentos similares ocorreram em todas as outras geometrias.

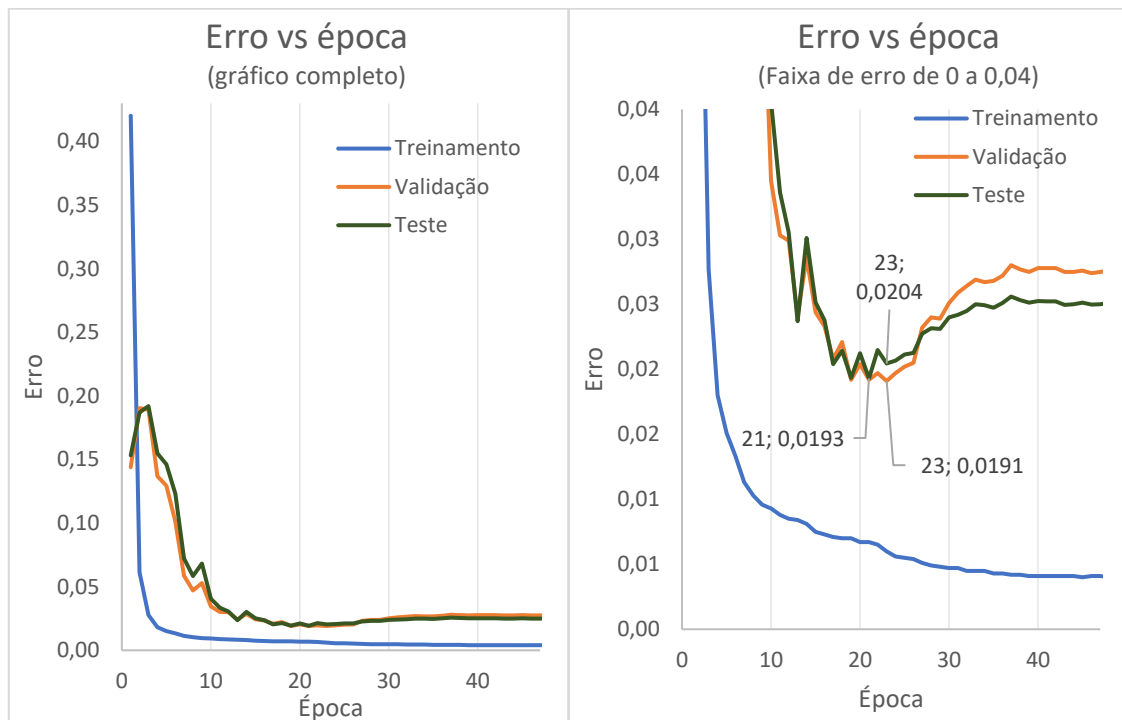
4.3 RESULTADOS RELACIONADOS AO TREINAMENTO DA REDE NEURAL

Nesta seção, são apresentados os resultados obtidos durante o treinamento da rede neural utilizada no projeto.

4.3.1 Análise do comportamento do erro

A Figura 32 apresenta os valores de erro (*BCE*) para os conjuntos de treinamento, validação e teste ao longo das 50 épocas de treinamento, permitindo identificar três fases distintas durante o processo de aprendizado da rede neural.

Figura 32 - Curvas de erro de treinamento, validação e teste versus época de treinamento



Fonte: Elaborada pelo Autor (2025).

Fase Inicial de Ajuste (Épocas 1 a 3): Inicialmente, houve uma queda acentuada no erro de treinamento, reduzindo de 0,4203 na primeira época para 0,0277 na terceira época. Simultaneamente, o erro de validação mostrou comportamento instável, partindo de 0,1440 e aumentando para 0,1904 na terceira época antes de reduzir para 0,1892 na terceira época. De maneira semelhante, o erro de teste, que partiu de 0,1534, aumentou até atingir um pico de 0,1920 na terceira época. Esse comportamento sugere uma rápida adaptação inicial do modelo aos dados de treinamento, mas também indica dificuldades iniciais em generalizar adequadamente.

Fase de Aprendizado Eficiente (Épocas 4 a 19): A partir da quarta época, o erro de treinamento continuou em redução constante, alcançando valores em torno de 0,007 na décima nona época. Paralelamente, os erros de validação e teste também diminuíram significativamente, atingindo mínimos de 0,0192 e 0,0193 respectivamente. Esta fase caracteriza-se por uma melhoria clara na generalização do modelo, demonstrando aprendizagem efetiva.

Fase de Estabilização e Sobreajuste (Épocas 20 a 50): Entre as épocas 19 e 22, o erro de validação não apresentou melhorias significativas, acionando assim o algoritmo de redução da taxa de aprendizagem descrita na seção 3.2.2. Essa estratégia fez com que o erro de validação alcançasse um valor mínimo de 0,0191 na época 23; entretanto, esse comportamento não foi acompanhado pelo erro de teste, que apresentou aumento nesse mesmo período, medindo 0,0204 na época 23. Após esse ponto, ambos os erros, validação e teste, apresentaram um aumento gradual até atingirem valores finais em torno de 0,0277 e 0,0251 respectivamente na última época. Este comportamento indica que o modelo começou a apresentar sinais de sobreajuste, perdendo gradualmente a capacidade de generalização aos novos dados, apesar das estratégias dinâmicas de redução da taxa de aprendizagem, que variaram de 10^{-4} no início do treinamento até 10^{-6} nas épocas finais.

4.3.2 Análise do comportamento do F1-score

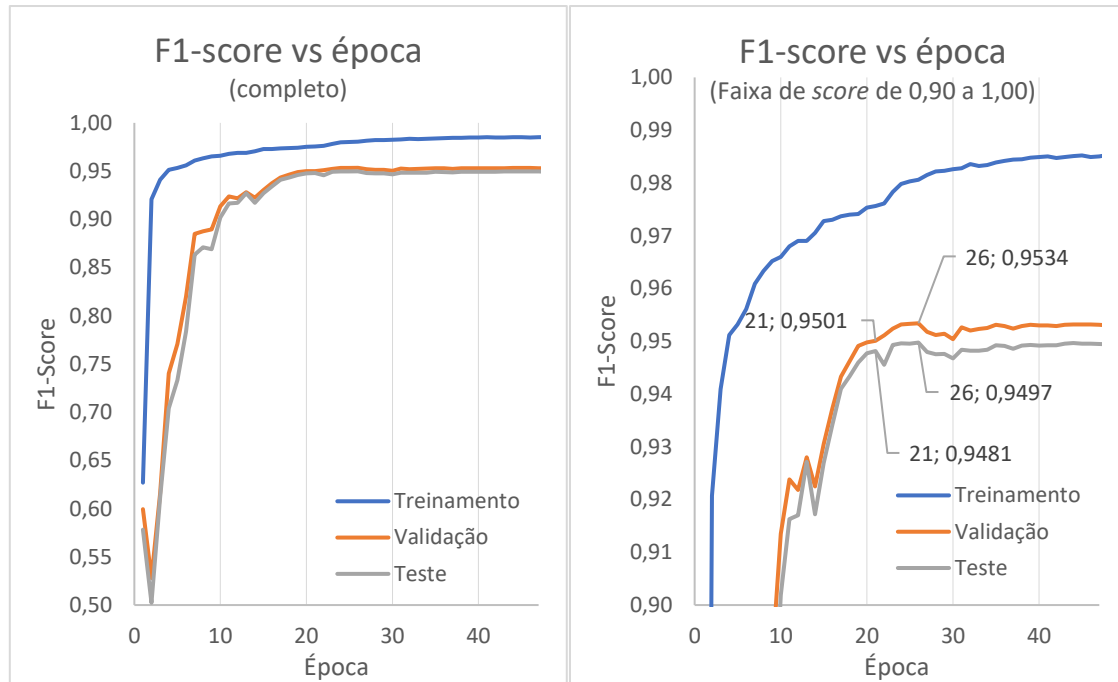
A análise dos valores de F1-score ao longo das 50 épocas revelou três fases distintas de desempenho da rede:

Fase Inicial de Ajuste (Épocas 1 a 3): Na fase inicial, observou-se um aumento rápido no F1-score de treinamento, que passou de 0,6269 para 0,9409 na terceira época. Entretanto, os conjuntos de validação e teste apresentaram comportamentos instáveis com valores iniciais de 0,5994 e 0,5782, reduzindo para 0,5279 e 0,5023 respectivamente na segunda época, e só então aumentando para 0,6156 e 0,6113 na terceira época. Essa fase apresenta uma aprendizagem inicial acelerada, porém com dificuldades claras de generalização.

Fase de Aprendizado Eficiente (Épocas 6 a 26): Durante esta fase, o F1-score continuou aumentando consistentemente em todos os conjuntos, de

treinamento, teste e validação. O F1-score do conjunto de treinamento atingiu 0,9806 na época 26, enquanto validação e teste alcançaram valores máximos de 0,9534 e 0,9497 respectivamente.

Figura 33 - Curva de F1-score ao longo das épocas



Fonte: Elaborada pelo Autor (2025).

Fase de Estabilização e Sobreajuste (Épocas 27 a 50): Após a época 26, observou-se um crescimento marginal do F1-score de treinamento, atingindo um valor máximo de 0,9853 nas épocas 48 e 49. Por outro lado, os valores para validação e teste apresentaram pequena redução e estabilização em torno de 0,9530 e 0,9493, respectivamente, sugerindo que o modelo pode estar atingindo um platô na capacidade de generalização e possivelmente iniciando um leve sobreajuste. Esses resultados ressaltam novamente a necessidade de estratégias adicionais para evitar perdas na capacidade generalização do modelo.

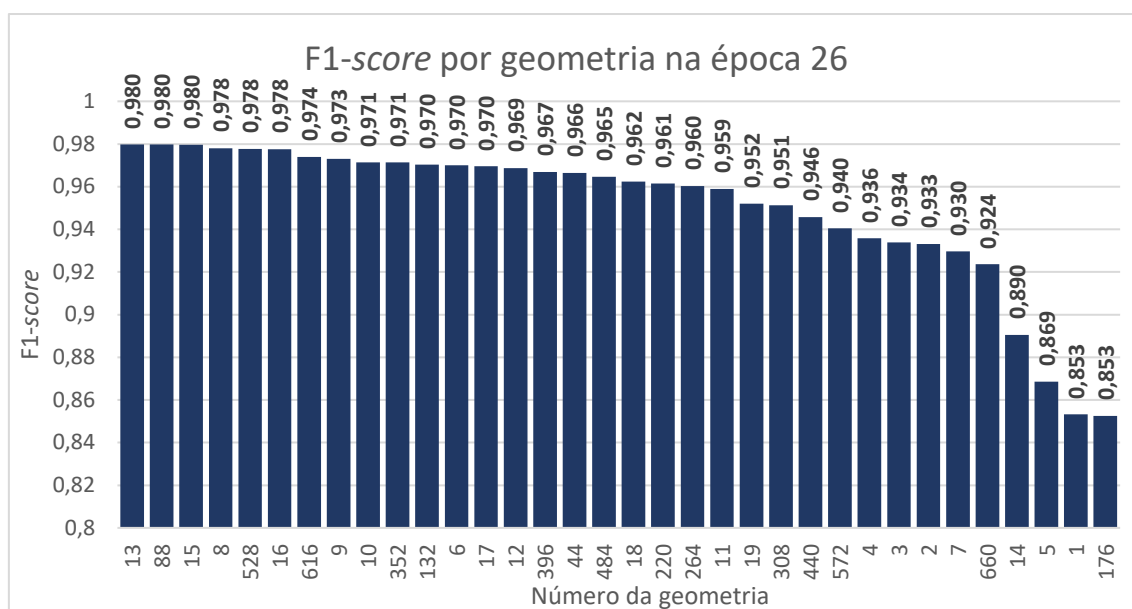
4.4 AVALIAÇÃO DA PERFORMANCE DA REDE NO CONJUNTO DE TESTES

As geometrias reservadas para teste foram então processadas pelo programa de cálculo das métricas e de tesselação para retornar as geometrias ao formato de malha, e assim possibilitar a avaliação dimensional.

4.4.1 Avaliação para seleção da melhor época

A partir da análise detalhada dos valores de erro (BCE) e F1-score, foi selecionada a época 26 como a melhor época para aplicação prática do modelo. Essa decisão baseou-se principalmente no desempenho superior do F1-score obtido nessa época, que atingiu o valor máximo médio no conjunto de testes (0,9497), indicando uma excelente capacidade do modelo de generalizar e realizar previsões robustas. Embora o erro nesta época não tenha sido o menor absoluto, manteve-se suficientemente baixo (aproximadamente 0,0212), confirmando um desempenho geral equilibrado e consistente da rede neural.

Figura 34 – F1-score por geometria na época 26



Fonte: Elaborada pelo Autor (2025).

O valor de F1-score obtido de 0,9497 representa uma média dos valores medidos para cada uma das geometrias de teste, de forma que algumas delas apresentem valores maiores que a média e outros abaixo dela. A Figura 34 mostra o F1-score medido para cada geometria usando a rede neural da época 26. Nela é possível perceber que existem variações significativas entre geometrias, refletidas em um desvio padrão calculado em 0,0346.

Entre os resultados obtidos, destacaram-se positivamente as geometrias 13, 88 e 15, que apresentaram os maiores valores de F1-score, próximos de

0,980, indicando uma excelente capacidade de previsão para esses casos específicos.

As geometrias que apresentaram resultados medianos, próximas ao valor médio, incluem os índices 19, 308 e 440, com valores de *F1-score* situados em torno de 0,95. Esses casos demonstram uma boa qualidade de previsão, porém com espaço para aperfeiçoamento das estratégias de treinamento ou ajustes da arquitetura.

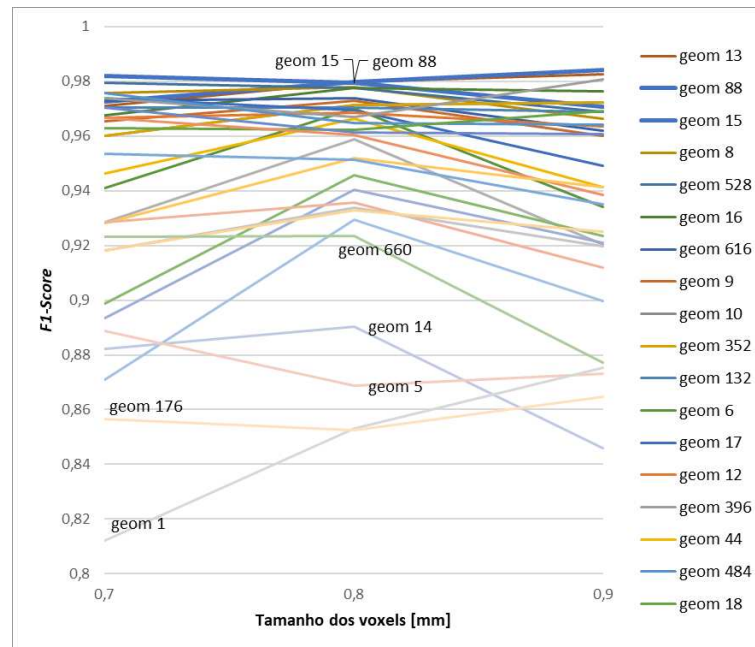
Por outro lado, algumas geometrias se destacaram negativamente, exibindo valores significativamente inferiores à média geral. Neste grupo estão as geometrias de índices 1, 5, 14 e 176, (sendo a geometria 176, uma cópia da geometria 5 rotacionada em 90 graus), com *F1-score* abaixo de 0,90, revelando claras dificuldades do modelo em prever corretamente essas geometrias específicas.

4.4.2 Impacto do Tamanho dos Voxels sobre o *F1-Score*

A Figura 35 apresenta os valores de *F1-score* obtidos por cada geometria do conjunto de testes ao serem avaliadas pelos modelos treinados com três diferentes níveis de voxelização: 0,7 mm, 0,8 mm e 0,9 mm. Cada linha no gráfico representa uma geometria distinta, permitindo a visualização direta da sensibilidade individual ao grau de refinamento adotado. Observa-se que, para a maioria das geometrias, o desempenho se mantém estável entre os diferentes níveis, com pequenas oscilações nos valores do *F1-score*.

No entanto, nota-se que geometrias específicas (como a 1, 5, 14 e 176) apresentaram desempenho consistentemente inferior em todos os refinamentos, sugerindo dificuldades de generalização do modelo para essas formas específicas. Por outro lado, ao se considerar o conjunto completo de geometrias, verificou-se que 50% dos casos obtiveram o maior *F1-score* com o modelo treinado com voxel de 0,8 mm, o que indica uma preferência estatística por esse nível de discretização. Isso sugere que o refinamento intermediário pode representar um ponto de equilíbrio entre resolução geométrica e estabilidade de treinamento.

Figura 35 – Gráfico do F1-score para cada geometria em diferentes graus de refinamento



4.4.3 Análise das geometrias tesselladas

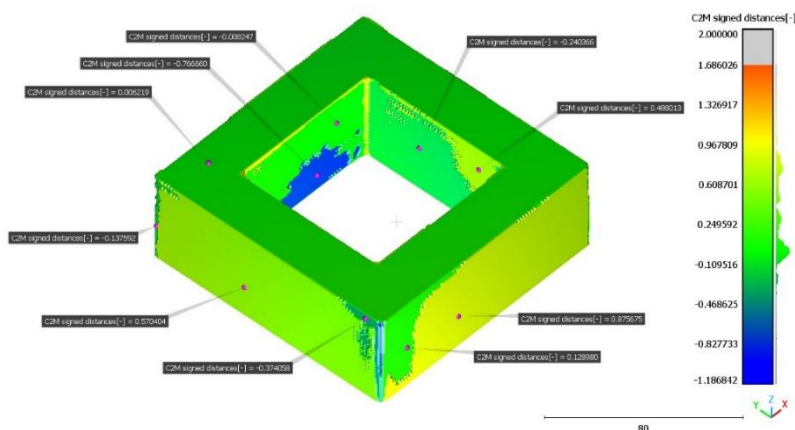
As imagens das geometrias tesselladas apresentadas a seguir oferecem uma melhor compreensão sobre o aprendizado da rede neural em capturar e reproduzir os comportamentos ensinados. As geometrias apresentadas nesta seção são algumas das que tiveram o melhor desempenho, quando avaliadas pelo F1-score, desempenho próximo à média dos dados e as geometrias que tiveram o pior desempenho, a fim de mostrar por completo os resultados da rede neural.

4.4.3.1 Geometria número 13

A Figura 36 mostra a geometria número 13 plotada com uma escala de cores que representa os desvios dimensionais em relação à geometria nominal. As regiões destacadas em vermelho representam áreas com excesso de material, indicando que a geometria predita possui dimensões maiores do que o esperado nestes locais, atingindo desvios máximos de até 1,686 mm. Os desvios dimensionais próximos a 1 mm são mostrados na cor amarela. As áreas verdes

são as regiões onde os resultados da rede neural são mais próximos da geometria nominal, com desvios entre +0,5 mm e -0,5 mm. Já as regiões azuis indicam locais com falta de material, ou seja, onde as dimensões da geometria predita estão abaixo da nominal em até -1,4257 mm neste caso.

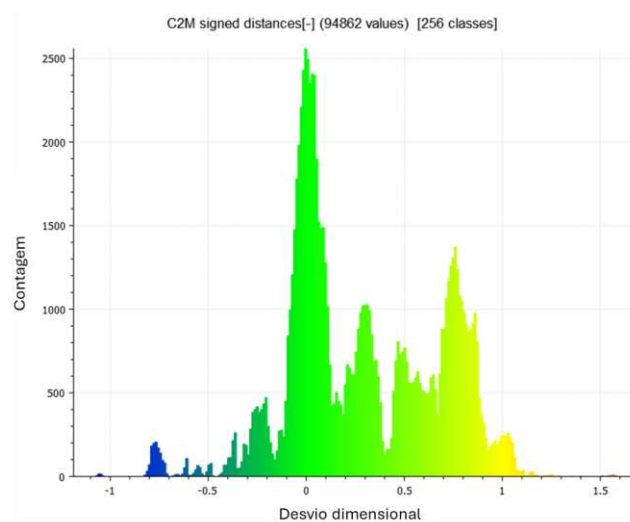
Figura 36 - Geometria número 13 calculada pela rede neural da época 26



Fonte: Elaborada pelo Autor (2025).

A análise do histograma dos desvios dimensionais, mostrado na Figura 37, revela que a maior parte das divergências se concentra na faixa entre -0,5 mm e +1,3 mm. Desvios superiores a esses valores são pontuais e podem ser considerados ruídos decorrentes do processo preditivo da rede neural.

Figura 37 – Histograma dos desvios dimensionais da geometria 13



Fonte: Elaborado pelo autor (2025).

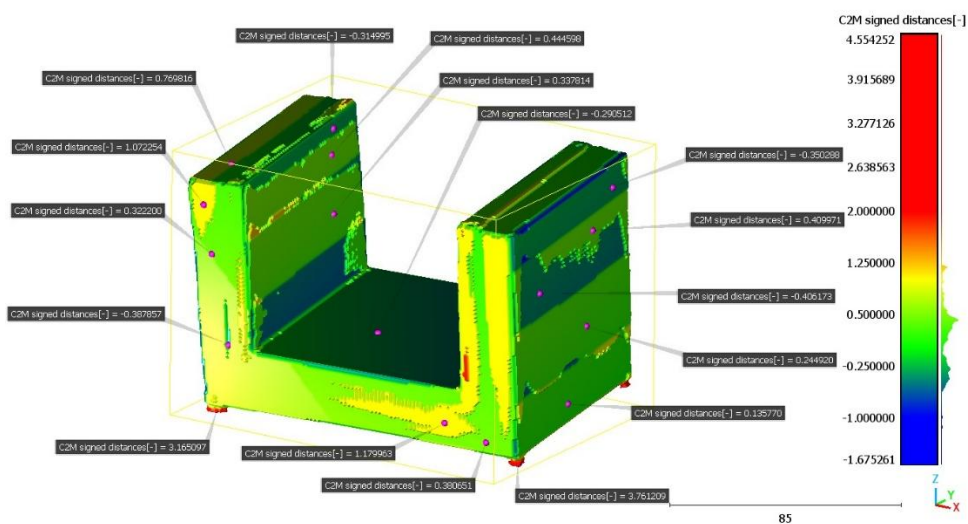
O objetivo original dessa geometria era avaliar a capacidade do modelo em reproduzir o comportamento da restrição da areia do molde sobre a contração do metal. Contudo, a geometria obtida da simulação convencional, após o processo de voxelização, apresentou apenas contração dimensional homogênea, sem alterações na forma, indicando que o modelo neural precisou prever apenas a contração e não prever nenhuma distorção na geometria.

4.4.3.2 Geometria número 15

A

Figura 38 apresenta a geometria número 15 predita pela rede neural, exibida com uma escala de cores que representa os desvios dimensionais em relação à geometria nominal. Essa geometria tinha como propósito capturar o efeito da restrição da areia durante a solidificação do metal, e este comportamento foi de fato observado: as paredes laterais da peça apresentam flexão para fora, coerente com a descrição da literatura sobre distorções em peças fundidas.

Figura 38 – Geometria número 15 calculada pela rede neural da época 26

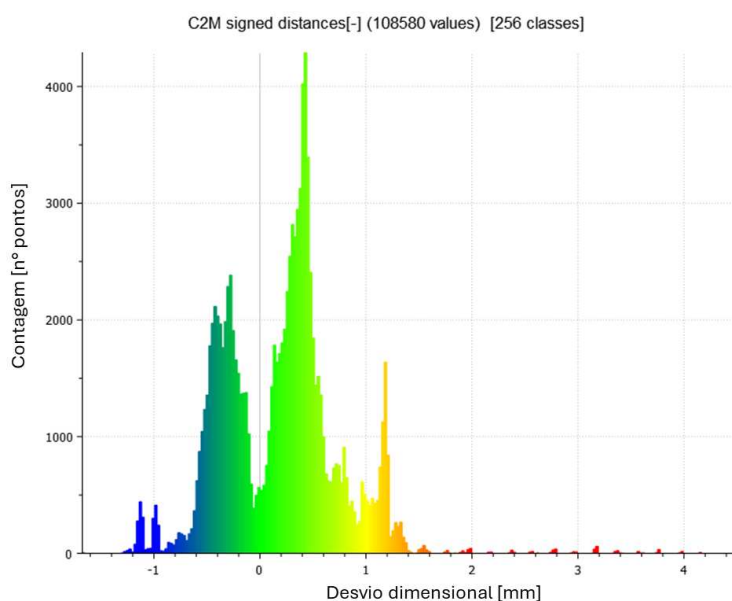


Fonte: Elaborado pelo autor (2025).

Quanto aos desvios locais, a maior variação dimensional positiva registrada foi de 4,5542 mm, ocorrendo em regiões claramente identificáveis como ruído, pois não apresentam correspondência com nenhum padrão geométrico típico do processo de fundição. Da mesma forma, o maior desvio negativo, de -1,6752 mm, também está localizado em uma área considerada como ruído. As demais distorções distribuem-se ao longo da peça com magnitudes mais moderadas e alinhadas ao comportamento físico esperado.

As dimensões globais da geometria predita, medidas próximas à base, foram de 89,6 mm × 132 mm, com altura de 89,6 mm. Em comparação, os valores nominais corretos são 88,8 mm × 132 mm, com altura de 88,8 mm. Observa-se, portanto, uma ótima precisão da rede neural em reproduzir o efeito da contração térmica.

Figura 39 – Histograma dos desvios dimensionais da geometria 15



Fonte: Elaborado pelo autor (2025).

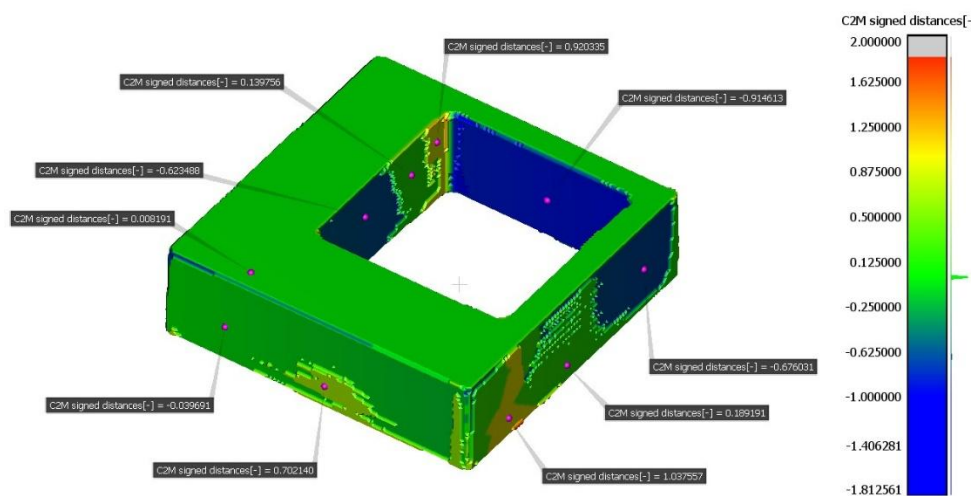
O histograma dos desvios dimensionais, mostrado na Figura 39, reforça essa análise, indicando que a maior parte dos pontos se encontram na faixa entre -0,6 mm e +1,2 mm. A distribuição é relativamente simétrica em torno do zero, com leve concentração em valores positivos, o que sugere uma tendência de acúmulo de material. As barras mais extremas — acima de 3 mm ou abaixo

de -1,5 mm — ocorrem com baixa frequência e reforçam a hipótese de ruído algorítmico, como também evidenciado na visualização espacial.

4.4.3.3 Geometria número 8

A Figura 40 apresenta a geometria número 8, predita pela rede neural, visualizada com mapeamento de cores que indicam os desvios dimensionais em relação à geometria nominal. O propósito dessa geometria era simular o efeito do resfriamento não uniforme sobre a dimensão e forma final da peça. Contudo, esse efeito foi bastante sutil no resultado da simulação convencional e, por isso, não se manteve representado na geometria após a voxelização. Assim, a geometria deformada considerava apenas o efeito da contração térmica homogênea.

Figura 40 – Geometria número 8 calculada pela rede neural da época 26

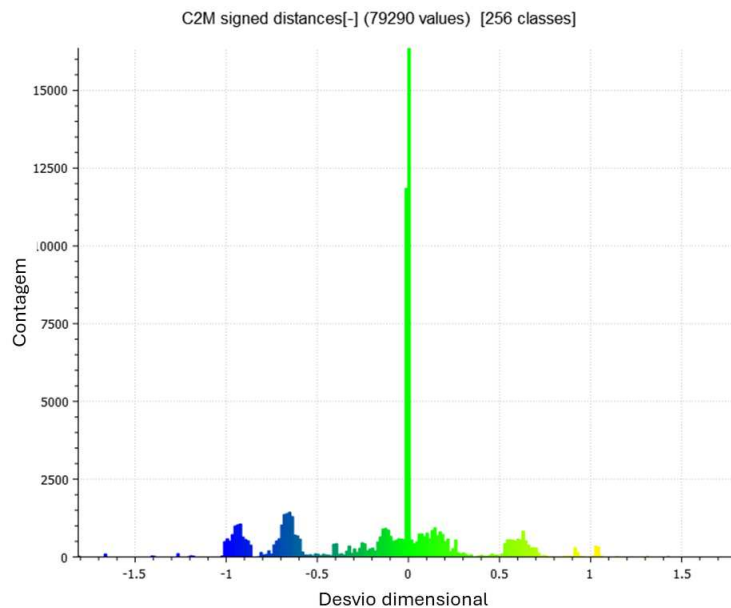


Fonte: Elaborado pelo autor (2025).

A análise dos desvios mostra que as discrepâncias mais relevantes ocorreram em regiões internas da peça, destacando-se os valores máximos positivos de 1,0376 mm e mínimos negativos de até -1,8126 mm. Os desvios negativos mais intensos concentram-se nas superfícies internas, indicando regiões com ausência de material. Embora os desvios estejam distribuídos ao longo da peça, eles não apresentam padrões geométricos que sugiram distorções físicas conhecidas, o que levanta a hipótese de que boa parte dessas variações possa estar associada a ruídos no processo preditivo.

As dimensões globais da geometria predita, como largura, comprimento e espessura foram calculadas pela rede neural com relativa precisão. As dimensões externas resultantes foram 155 x 156 x 45mm e quando comparadas com as dimensões esperadas 155 x 155 x 45mm é possível concluir que a rede neural efetuou a previsão da contração térmica com pouco erro de forma geral.

Figura 41 – Histograma dos desvios dimensionais da geometria 8



Fonte: Elaborado pelo autor (2025).

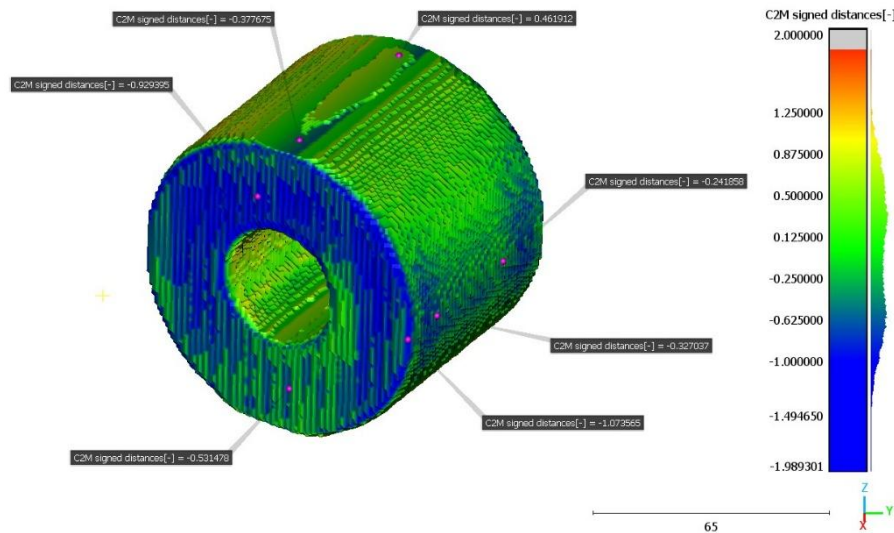
A Figura 41 que apresenta o histograma dos desvios dimensionais, confirma esse comportamento. Nota-se um pico acentuado em torno de zero, indicando um alto número de pontos com pequeno desvio. A maior parte das variações está contida entre -1,0 mm e +0,7 mm, com predomínio de desvios negativos. Os valores nas extremidades do histograma, embora presentes, são pouco representativos em termos estatísticos e provavelmente correspondem a ruídos.

4.4.3.4 Geometria número 132

A Figura 42 apresenta a geometria número 132 predita pela rede neural, exibida com coloração baseada nos desvios dimensionais em relação à geometria nominal. O objetivo desta geometria era simular os efeitos da dilatação da areia sobre a forma final da peça. No entanto, como esse fenômeno não foi

incluído na simulação convencional, a geometria esperada contemplava apenas a contração térmica do metal.

Figura 42 - Geometria número 132 calculada pela rede neural da época 26



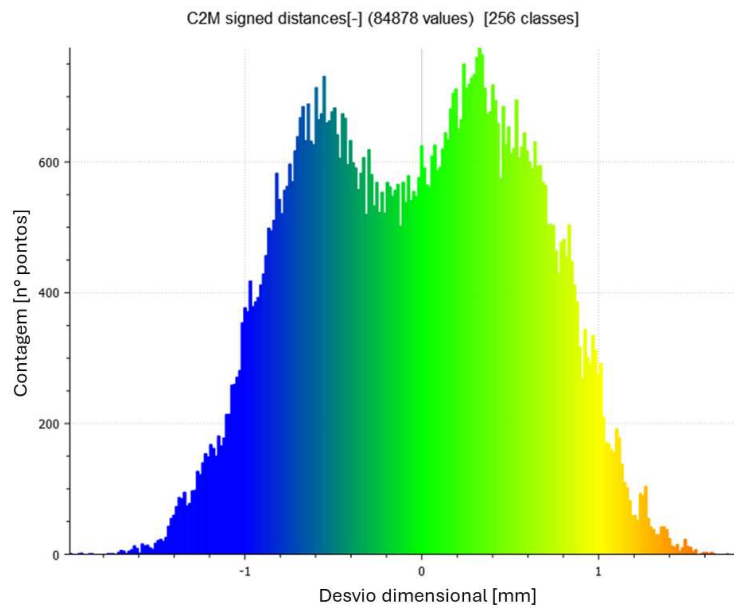
Fonte: Elaborado pelo autor (2025).

O formato cilíndrico da peça gerou dificuldades na representação voxelizada, aliando a rotação da peça 60 graus em relação ao sistema cartesiano e conseqüentemente à grade de voxels. Isso é evidenciado pela superfície serrilhada e pela densidade irregular de pontos com maiores desvios em regiões inclinadas, onde a grade cartesiana não representa com precisão a curvatura.

Os valores extremos de desvio variam de -1,9893 mm até aproximadamente +2 mm. As regiões com maiores desvios negativos concentram-se nas superfícies internas, especialmente nas áreas cilíndricas voltadas para o centro da peça, sugerindo ausência de material — coerente com uma representação subamostrada ou erro de discretização. As áreas externas, por outro lado, apresentam desvios levemente positivos, em sua maioria inferiores a 1 mm.

As dimensões globais da peça predita foram de 88 mm × 109 mm × 111 mm, enquanto a geometria nominal esperada possuía 89,63 mm × 111 mm × 111 mm. Indicando que houve desvio na previsão do comprimento da peça e largura, principalmente.

Figura 43 - Histograma dos desvios dimensionais da geometria 132



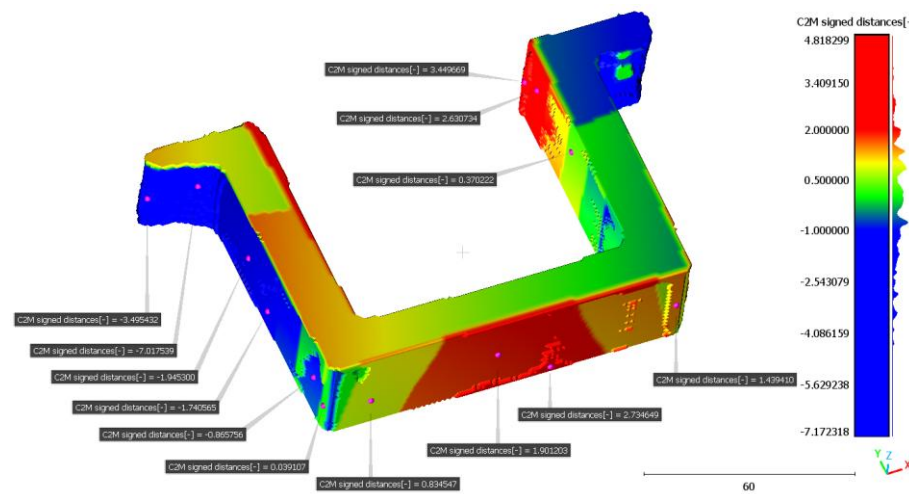
Fonte: Elaborado pelo autor (2025).

A Figura 39 apresenta o histograma dos desvios dimensionais. Observa-se uma distribuição em torno de zero, com leve assimetria — sendo o pico positivo mais pronunciado. A maior parte dos desvios está contida entre -1,2 mm e +1,2 mm, o que é coerente com a expectativa de contração térmica em geometrias com curvatura.

4.4.3.5 Geometria número 5

A Figura 44 apresenta a geometria número 5, cujo propósito era simular os efeitos da restrição da areia na forma e nas dimensões da peça fundida. A geometria esperada, baseada na simulação convencional, exibe paredes laterais fortemente curvadas para fora, além de uma aba fina localizada nas extremidades superiores, representando um desafio adicional de modelagem.

Figura 44 - Geometria número 5 calculada pela rede neural da época 26

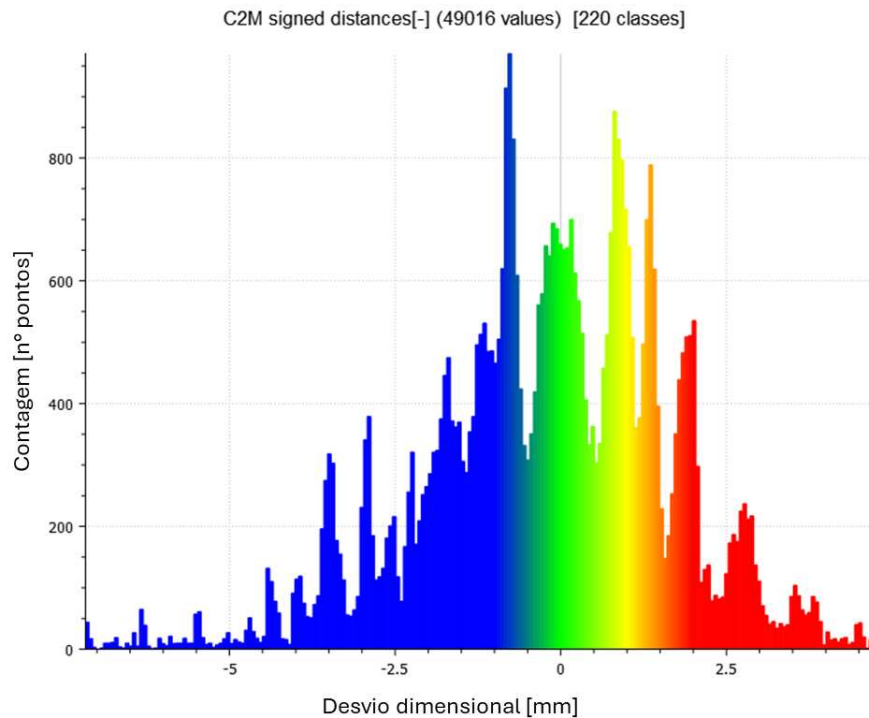


Fonte: Elaborado pelo autor (2025).

A previsão da rede neural mostra, de forma geral, que o padrão de distorção das paredes foi parcialmente assimilado. As curvaturas externas estão presentes e os desvios positivos (acúmulo de material) atingem até +4,8183 mm, sobretudo nas regiões superiores das paredes. No entanto, o modelo demonstrou notável dificuldade em representar corretamente a aba fina localizada nas extremidades, onde surgem desvios negativos intensos que chegam a -7,1723 mm. A geometria nessas regiões se rompe ou desaparece parcialmente, indicando que o modelo não conseguiu preservar estruturas delicadas e localizadas.

A Figura 45, correspondente ao histograma dos desvios, confirma esse comportamento e exibe um comportamento assimétrico, com pico em torno de zero, mas com dispersão expressiva tanto para o lado positivo quanto negativo. Há uma quantidade relevante de pontos com desvios superiores a +2 mm e também uma população significativa com erros abaixo de -4 mm, associados às falhas de representação da aba.

Figura 45 - Histograma dos desvios dimensionais da geometria 5



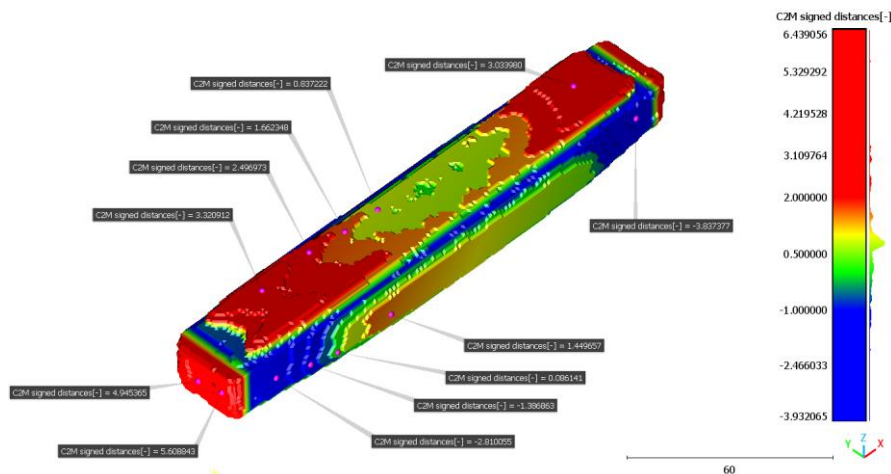
Fonte: Elaborado pelo autor (2025).

Em síntese, a geometria 5 demonstra que o modelo neural foi capaz de assimilar e reproduzir o padrão de distorção das paredes causado pela restrição da areia. No entanto, o desempenho do modelo foi comprometido em regiões com geometria complexa e descontinuidade, como a aba, evidenciando limitações na generalização espacial para elementos com variações abruptas ou pouco representados no conjunto de treinamento.

4.4.3.6 Geometria número 1

A Figura 46 apresenta a geometria número 1, uma das mais simples do conjunto de testes, composta por uma barra prismática sem detalhes ou saliências. O objetivo dessa geometria era reproduzir exclusivamente a contração térmica homogênea do metal, sem a influência de restrições da areia do molde. Por ser uma situação física básica e uma forma geométrica trivial, esperava-se uma previsão precisa por parte da rede neural.

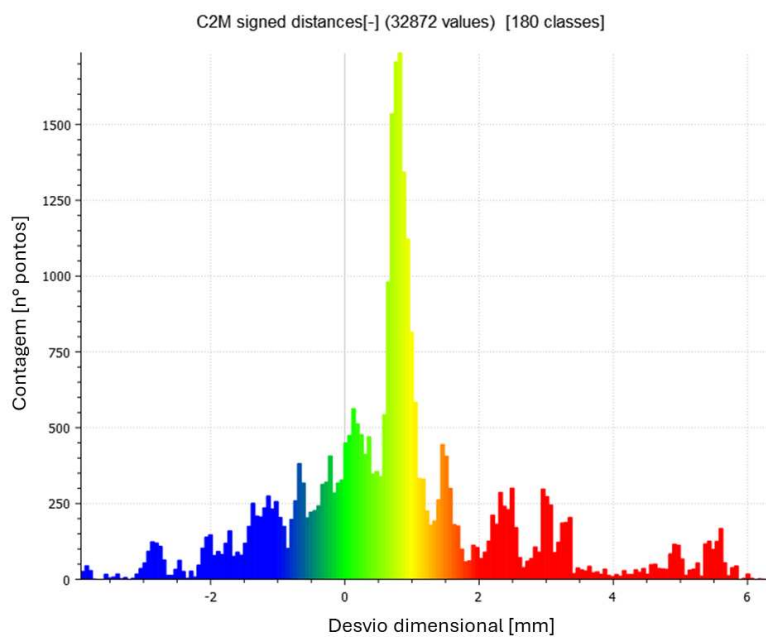
Figura 46 - Geometria número 1 calculada pela rede neural da época 26



Fonte: Elaborado pelo autor (2025).

No entanto, os resultados indicam o oposto. A geometria número 1 apresentou o pior desempenho entre todas as testadas, com um F1-score de apenas 0,853. A visualização mostra grandes regiões com desvios significativos. Nas extremidades da barra observa-se acúmulo de material com desvios positivos que chegam a +6,4391 mm, enquanto as laterais centrais mostram retrações de até -3,8374 mm. Este comportamento distorce totalmente a expectativa de contração uniforme.

Figura 47 - Histograma dos desvios dimensionais da geometria 1



Fonte: Elaborado pelo autor (2025).

O histograma de desvios dimensionais, mostrado na Figura 47, revela um padrão de distribuição irregular, com uma grande concentração de desvios positivos entre +1 e +4 mm e uma cauda negativa que alcança -3,9 mm. Isso evidencia um excesso de material distribuído ao longo da face superior da barra, provavelmente decorrente de um erro sistemático de superestimação da geometria durante a previsão.

A forma global da peça se manteve linear, mas o modelo falhou ao distribuir corretamente o volume ao longo do eixo longitudinal, resultando em um volume excessivo nas extremidades e regiões intermediárias. A rede neural demonstrou baixa capacidade de generalização para esse caso puramente contrátil e simétrico, o que sugere que o modelo ainda apresenta dificuldade para lidar com geometrias simples sem elementos de restrição que orientem o aprendizado espacial.

4.4.4 Utilização de recursos computacionais – Simulação convencional

A avaliação do consumo de recursos computacionais foi realizada tanto para o método convencional de simulação, utilizando o software MAGMASOFT, quanto para a abordagem baseada em redes neurais tridimensionais.

Na simulação convencional, foram processadas as 19 geometrias, cada uma submetida a duas etapas: resfriamento e análise de tensões mecânicas, totalizando 38 simulações. O tempo total acumulado foi de 18,5 horas, com um tempo médio de 4 minutos por geometria na fase de resfriamento e 58 minutos na fase de tensões, resultando em um tempo médio total de 62 minutos por geometria.

Tabela 3 – Medição dos recursos computacionais utilizados na simulação

Tempo total de simulação	19,6 h
Tempo médio (resfriamento)	4 min
Tempo médio (tensões)	58 min
Tempo médio (por geometria)	62 min
Memória média (resfriamento)	171 mb
Memória média (tensões)	278 mb

Fonte: Elaborada pelo autor (2025).

O uso de memória RAM também variou entre as etapas. Durante a simulação de resfriamento, a memória residente média foi de 171 MB, enquanto na etapa de tensões o consumo médio foi de 278 MB. Essas informações foram extraídas diretamente dos arquivos de log gerados pelo MAGMASOFT, com base nas entradas *CPU usage* e *memory usage*.

4.4.5 Utilização de recursos computacionais – Redes neurais

O treinamento da rede neural foi realizado no ambiente de programação Colab utilizando a GPU A100, conforme descrito na seção 3.2.5. O processo completo das 50 épocas de treinamento teve duração total de aproximadamente 3 horas e 20 minutos (11.979 segundos). Durante esse período, a memória de GPU foi ocupada de forma constante em 36,69 GB, enquanto a memória RAM da CPU apresentou crescimento progressivo, atingindo um pico de 11,75 GB ao final do treinamento.

Após o término do treinamento, o modelo resultante foi utilizado para realizar inferências sobre novas geometrias. O tempo de processamento por geometria foi de 1 segundo, com consumo de memória significativamente inferior ao exigido durante a fase de treinamento. Como o treinamento é realizado apenas uma vez, o custo computacional por peça na fase de aplicação (inferência) torna-se praticamente irrelevante, especialmente quando comparado à abordagem convencional baseada em simulações físicas.

5 CONCLUSÕES

5.1 DESEMPENHO GERAL DA REDE NEURAL

Este trabalho demonstrou a viabilidade da aplicação de redes neurais convolucionais tridimensionais na previsão de deformações geométricas resultantes de processos de fundição. A abordagem desenvolvida foi capaz de aprender padrões complexos de distorção a partir de geometrias voxelizadas, atingindo um F1-score médio de 0,9497 no conjunto de testes. Esse resultado valida a estratégia de aprendizado supervisionado com dados simulados, demonstrando o potencial da inteligência artificial como ferramenta de suporte ao planejamento dimensional de peças fundidas.

A análise das curvas de erro e F1-score revelou que a rede neural apresentou boa capacidade de generalização até a época 26, ponto no qual foi atingido o melhor desempenho. A partir dessa época, foram observados indícios de sobreajuste, evidenciando a importância de técnicas de regularização e controle dinâmico da taxa de aprendizagem para preservar o equilíbrio entre aprendizado e generalização.

5.2 LIMITAÇÕES IDENTIFICADAS

A avaliação individualizada das geometrias do conjunto de testes possibilitou a identificação de pontos críticos no desempenho do modelo. Geometrias como as de índices 1, 5, 14 e 176 apresentaram valores de F1-score sistematicamente inferiores, mesmo com diferentes graus de refinamento. Esses resultados revelam dificuldades do modelo em lidar com formas geométricas específicas.

Essas limitações indicam que, embora a rede neural tenha aprendido a generalizar padrões comuns de deformação, sua capacidade de extrapolação para alguns casos foi restrita. A inclusão de mais exemplos representativos dessas condições críticas poderá aprimorar significativamente a robustez do modelo.

5.3 INFLUÊNCIA DO GRAU DE REFINAMENTO

A análise de sensibilidade ao tamanho dos voxels no conjunto de treinamento revelou que o valor de 0,8 mm proporcionou os melhores resultados para a maioria das geometrias avaliadas. Essa constatação sugere que esse nível de discretização representa um ponto ótimo de equilíbrio entre a fidelidade geométrica e a estabilidade da rede durante o treinamento.

Refinamentos mais altos (0,7 mm) podem capturar mais detalhes geométricos, porém também introduzem ruído e aumentam a complexidade do problema. Por outro lado, voxelizações mais grosseiras (0,9 mm) tendem a comprometer a representação de elementos importantes da geometria.

5.4 COMPARAÇÃO COM A SIMULAÇÃO CONVENCIONAL

A abordagem baseada em redes neurais também demonstrou vantagens significativas quando comparada ao método convencional de simulação com o software MAGMASOFT. Enquanto o tempo médio por simulação convencional ultrapassou uma hora por geometria, a previsão com a rede neural foi realizada em questão de segundos. Isso representa um ganho substancial em tempo de resposta, especialmente relevante em contextos industriais que demandam a análise de múltiplas geometrias em curto prazo.

Além da velocidade, a inferência com a rede neural requer significativamente menos recursos computacionais, podendo ser realizada com menor uso de memória e processamento, o que amplia sua aplicabilidade em ambientes com infraestrutura limitada. No entanto, é importante destacar que as vantagens observadas da rede neural se referem especificamente à previsão de alterações dimensionais. O MAGMASOFT, por sua vez, é um software de simulação extremamente completo, capaz de modelar uma ampla gama de fenômenos físicos associados ao processo de fundição, como transferência de calor, solidificação, tensões mecânicas, fluxo do metal, formação de defeitos, entre outros. Dessa forma, a rede neural estudada apresenta vantagens pontuais em relação ao tempo de processamento, mas não substitui o escopo abrangente de análise fornecido por ferramentas de simulação física como o MAGMASOFT.

5.5 REQUISITOS PARA APLICAÇÃO INDUSTRIAL DO SOFTWARE

Apesar dos resultados promissores obtidos neste trabalho, algumas adaptações são necessárias para viabilizar o uso da ferramenta em contextos industriais reais. A primeira limitação diz respeito à resolução espacial. O refinamento atual, com voxels de 0,8 mm, é suficiente para demonstrar a viabilidade da abordagem, porém não atende às exigências dimensionais de precisão de muitas peças na indústria. Para que a rede neural seja capaz de descrever e prever alterações dimensionais nesta escala, seria necessário reduzir significativamente o tamanho dos voxels, idealmente para valores em torno de 0,1 mm. Tal medida aumentaria a resolução da geometria de entrada e, conseqüentemente, a capacidade da rede de identificar padrões sutis de deformação.

Além disso, seria necessário ampliar o volume máximo das geometrias processadas. O presente modelo foi limitado a peças de até 200 × 200 × 100 mm, dimensão considerada reduzida para processos de fundição em areia, os quais frequentemente envolvem peças com mais de um metro de comprimento. Tanto a elevação da resolução quanto a ampliação do volume exigiriam mudanças significativas na arquitetura da rede neural, tornando-a mais profunda ou utilizando estruturas que melhor suportem a alta complexidade dos dados de entrada. Essas alterações também implicariam aumento no consumo de memória da GPU, maior tempo de treinamento e possível necessidade de hardware mais robusto.

Outro aprimoramento relevante seria substituir os dados simulados utilizados no treinamento por geometrias obtidas a partir de corpos de prova reais fundidos em moldes de areia e digitalizados via escaneamento tridimensional. Como o programa já opera com arquivos no formato STL, essa substituição pode ser feita de forma direta, permitindo utilizar geometrias reais deformadas como dados de entrada no processo de treinamento. Dessa forma, a base de dados refletiria integralmente os comportamentos físicos, térmicos e estruturais envolvidos no processo de fundição, elevando substancialmente a confiabilidade do modelo.

REFERÊNCIAS

ARAFIN, Palisa; BILLAH, AHM Muntasir; ISSA, Anas. **Deep learning-based concrete defects classification and detection using semantic segmentation.** *Structural Health Monitoring*, vol. 23, n. 1, p. 383–409, jan. 2024. Disponível em: <https://doi.org/10.1177/14759217231168212>. Acesso em: 28 de março de 2025.

BAGHANI, Amir; DAVAMI, Parviz; VARAHRAM, Naser; SHABANI, Mohsen Ostad. **Investigation on the effect of mold constraints and cooling rate on residual stress during the sand-casting process of 1086 steel by employing a thermomechanical model.** *Metallurgical and Materials Transactions B*, vol. 45, n. 4, p. 1157–1169, jun. 2014. DOI: [10.1007/s11663-013-0015-6](https://doi.org/10.1007/s11663-013-0015-6)

BATHE, Klaus-Jürgen. **Finite Element Procedures in Engineering Analysis.** Englewood Cliffs, NJ: Prentice Hall, 1996. xiv, 1037 p. ISBN 0-13-301458-4.

BERTELLI, Felipe. **Modelagem numérica e análise experimental de parâmetros térmicos e microestruturais na solidificação radial de ligas binárias.** 2012. - UNICAMP, Campinas, 2012.

CAMPBELL, John. **Complete Casting Handbook: Metal Casting Processes, Techniques and Design.** 1. ed. Oxford: Butterworth-Heinemann, 2011. 1220 p. ISBN 978-1-85617-809-9.

CHEN, Zhengyang; LI, Yueyang; ZHAO, Fengrui; LI, Shi; ZHANG, Ji. **Progress in numerical simulation of casting process.** *Measurement and Control*, v. 55, n. 5–6, p. 257–264, maio 2022. DOI: 10.1177/00202940221102656.

CHOLLET, François. **Deep Learning with Python.** Segunda edição. Shelter Island, NY: Manning Publications, 2021.

ÇIÇEK, Özgün; ABDULKADIR, Ahmed; LIENKAMP, Soeren S.; BROX, Thomas; RONNEBERGER, Olaf. **3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation**. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Cham: Springer, 2016. p. 424–432.

DABLAIN, Damien; JACOBSON, Kristen N.; BELLINGER, Colin; ROBERTS, Mark; CHAWLA, Nitesh V. **Understanding CNN fragility when learning with imbalanced data**. *Machine Learning*, [S. l.], v. 112, p. 4785–4810, abr. 2023. DOI: 10.1007/s10994-023-06326-9.

DOWLING, Norman E. **Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue**. 4th. ed. Pearson Education, 2012.

EDACHERIAN, Abhilash; ABHILASH, E; JOSEPH, M A. **Modelling and Simulation of Casting Process: an Overview**. *Indian Foundry Journal*, v. 55, n. 10, p. 28–37, 2009. Disponível em: <https://www.researchgate.net/publication/270757498>.

EIRIKSSON, Eyþór Rúnar. **Computer Vision for Additive Manufacturing**. APA, 2018. Disponível em: www.compute.dtu.dk.

ERTEL, Wolfgang. **Introduction to Artificial Intelligence**. 2 ed. Cham: Springer, 2017. 370 p. (Undergraduate Topics in Computer Science). ISBN 978-3-319-58486-7

GALLES, D.; BECKERMANN, C. **Effect of sand dilation on core expansion during steel casting**. In: , 2015. IOP Conference Series: Materials Science and Engineering, 2015.

GALLES, D.; BECKERMANN, C. **Effect of sand dilation on distortions and pattern allowances during steel sand casting**. *International Journal of Cast Metals Research*, v. 30, n. 5, p. 257–275, 2017a.

GALLES, D.; BECKERMANN, C. **Modeling of distortion of a steel bracket sand casting**. In: , 2016. TMS Annual Meeting, 2016. p. 43–50.

GALLES, D.; BECKERMANN, C. **Prediction of distortions and pattern allowances during sand casting of a steel bracket.** *International Journal of Cast Metals Research*, v. 30, n. 3, p. 133–147, 2017b.

GALLES, Daniel J.; HARDIN, Richard A.; BECKERMANN, Christoph. **Measurement and simulation of deformation and stresses in steel casting.** In: *Modeling of Casting, Welding, and Advanced Solidification Processes XII*. Schladming, Áustria, 17–22 jun. 2012. In: *IOP Conference Series: Materials Science and Engineering*, v. 33, n. 1, p. 012049, 2012. DOI: 10.1088/1757-899X/33/1/012049.

GÉRON, Aurélien. **Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems.** 2ª edição. O'Reilly Media, 2019.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning.** Primeira edição. Cambridge, Massachusetts: The MIT Press, 2016.

HATTEL, J H. SOME FUNDAMENTALS OF MODELLING OF CASTING PROCESSES. **Dansk Metallurgisk Selskab**, p. 87–100, 2005.

INCROPERA, Frank P.; DEWITT, David P.; BERGMAN, Theodore L.; LAVINE, Adrienne S. **Fundamentos de transferência de calor e de massa.** Tradução de Eduardo Mach Queiroz; Fernando Luiz Pellegrini Pessoa. 6. ed. São Paulo: LTC, 2007.

INOUE, Y. *et al.* **Effect of sand mold models on the simulated mold restraint force and the contraction of the casting during cooling in green sand molds.** *Journal of Materials Processing Technology*, v. 213, n. 7, p. 1157–1165, 2013.

IOFFE, Sergey; SZEGEDY, Christian. **Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.** In: *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*. Lille, França, 6–11 jul. 2015. In: *Proceedings of Machine Learning*

Research, [S. l.], v. 37, p. 448–456, 2015. Disponível em: <http://arxiv.org/abs/1502.03167>, acesso em 28 de março de 2025.

KOTAS, Petr. **Integrated Modeling of Process, Structures and Performance in Cast Parts**. 2011. - Technical University of Denmark, 2011.

LOSHCHILOV, Ilya; HUTTER, Frank. **Decoupled weight decay regularization**. *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. Nova Orleans, EUA, 6–9 maio 2019. Disponível em: <http://arxiv.org/abs/1711.05101>, acesso em 28 de março de 2025.

NASIRI, Sara; HELSPER, Julien; JUNG, Matthias; FATHI, Madjid. **DePicT Melanoma Deep-CLASS: a deep convolutional neural networks approach to classify skin lesion images**. *BMC Bioinformatics*, v. 21, supl. 2, art. 84, 2020. DOI: 10.1186/s12859-020-3351-y.

PINTELAS, Emmanuel; LIVIERIS, Ioannis E.; PINTELAS, Panagiotis E. **A convolutional autoencoder topology for classification in high-dimensional noisy image datasets**. *Sensors*, v. 21, n. 22, 2021.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. **U-Net: Convolutional networks for biomedical image segmentation**. *In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. p. 234–241. Disponível em: <http://arxiv.org/abs/1505.04597>, acesso em 28 de março de 2025.

SAADA, R. Ami; BONNET, G.; BOUVARD, D. **Thermomechanical behavior of casting sands: experiments and elastoplastic modeling**. *International Journal of Plasticity*, [S. l.], v. 12, n. 3, p. 273–294, 1996. DOI: 10.1016/0749-6419(96)00003-3.

SHELHAMER, Evan; LONG, Jonathan; DARRELL, Trevor. **Fully convolutional networks for semantic segmentation**. *In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 2015. p. 3431–3440. Disponível em: <http://arxiv.org/abs/1605.06211>, acesso em 28 de março de 2025.

Shen, Z., Shang, X., Zhao, M., Dong, D., & Xu, B. (2019). **A learning-based framework for error compensation** in 3D printing. *IEEE Transactions on Cybernetics*. doi:10.1109/TCYB.2018.2879923

SURESH, S. **Fatigue of materials**. 2nd. ed. Cambridge University Press, 2004.

SVIDRÓ, Judit; DIÓSZEGI, Attila; SVIDRÓ, József Tamás. **The origin of thermal expansion differences in various size fractions of silica sand**. *International Journal of Cast Metals Research*, v. 33, n. 6, p. 242–249, 2020.

TAHA, Abdel Aziz; HANBURY, Allan. **Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool**. *BMC Medical Imaging*, v. 15, n. 1, 2015.

TENSORFLOW. **BinaryCrossentropy - TensorFlow API Docs**, 2024a. Disponível em: https://www.tensorflow.org/api_docs/python/tf/keras/losses/BinaryCrossentropy. Acesso em: 29 mar. 2025.

TENSORFLOW. **Tf.keras.layers.BatchNormalization**, 2024b. Disponível em: https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization. Acesso em: 29 mar. 2025.

THIEL, Jerry; RAVI, Sairam. **Predicting Casting Dimensions With Computer Process Modeling**. *Modern Casting*, v. 106, n. 7, p. 28–31, 2016. Disponível em: <https://www.moderncasting.com/articles/2016/07/11/predicting-casting-dimensions-computer-process-modeling>. Acesso em: 14 nov. 2024.

TRIMESH. **Trimesh.voxel.creation**, 2022. Disponível em: <https://trimesh.org/trimesh.voxel.creation.html#trimesh.voxel.creation.voxelize>. Acesso em: 29 mar. 2025.

VERRAN, Guilherme O.; REBELLO, M. A.; OLIVEIRA, C. G. **INFLUÊNCIA DA VELOCIDADE DE ENCHIMENTO SOBRE A FORMAÇÃO DE DEFEITOS EM PEÇAS DE ALUMÍNIO: SIMULAÇÃO NUMÉRICA VS.**

RESULTADOS EXPERIMENTAIS. Congresso Brasileiro de Engenharia e Ciência dos Materiais, n. 14, p. 42201–42210, 2000.

WANG, Yueqing; YAO, Hong; ZHAO, Shen. **An efficient and effective convolutional auto-encoder extreme learning machine network for 3D feature learning.** *Neurocomputing*, v. 174, p. 988-998, 2016. doi:10.1016/j.neucom.2015.10.035.

Wang, Y., Yao, H., & Zhao, S. **An efficient and effective convolutional auto-encoder extreme learning machine network for 3D feature learning.** *Neurocomputing*, 2016. <https://doi.org/10.1016/j.neucom.2015.10.035>

WARDE-FARLEY, David; GOODFELLOW, Ian J.; COURVILLE, Aaron. **An empirical analysis of dropout in piecewise linear networks**, 2013. Disponível em: <http://arxiv.org/abs/1312.6197>, acesso em 28 de março de 2025.

ZHANG, Junzhe; RAJAN, Kaushik; NANDY, Siddharth. **Efficient Memory Management for GPU-based Deep Learning Systems**, 2019. Disponível em: <http://arxiv.org/abs/1903.06631>.

ZHAO, Meihua; WANG, Tao; LIU, Changqing. **Nonlinear deformation prediction and compensation for 3D printing based on CAE neural networks.** In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION SCIENCE AND ENGINEERING, CASE, 2019, Vancouver: IEEE Computer Society, 2019. p. 667-672. Disponível em: <https://doi.org/10.1109/COASE.2019.8842885>.

ZIENKIEWICZ, O C; TAYLOR, R L; ZHU, JZ. **The Finite Element Method: Its Basis and Fundamentals, Sixth Edition.** 6th. ed. Burlington, MA: Elsevier Butterworth-Heinemann, 2005.

APÊNDICE A – PROPRIEDADES MECÂNICAS E FÍSICAS DO METAL UTILIZADO

Tabela dos parâmetros gerais do material utilizado (GJL-250)

Material: GJL-250

General Parameters

Material type:

Solidus temperature: °C

Liquidus temperature: °C

Niyama Criterion Temperature: °C

Thermo Criteria Temperature: °C

Initial temperature: °C

Latent heat: kJ/kg

Solidification morphology:

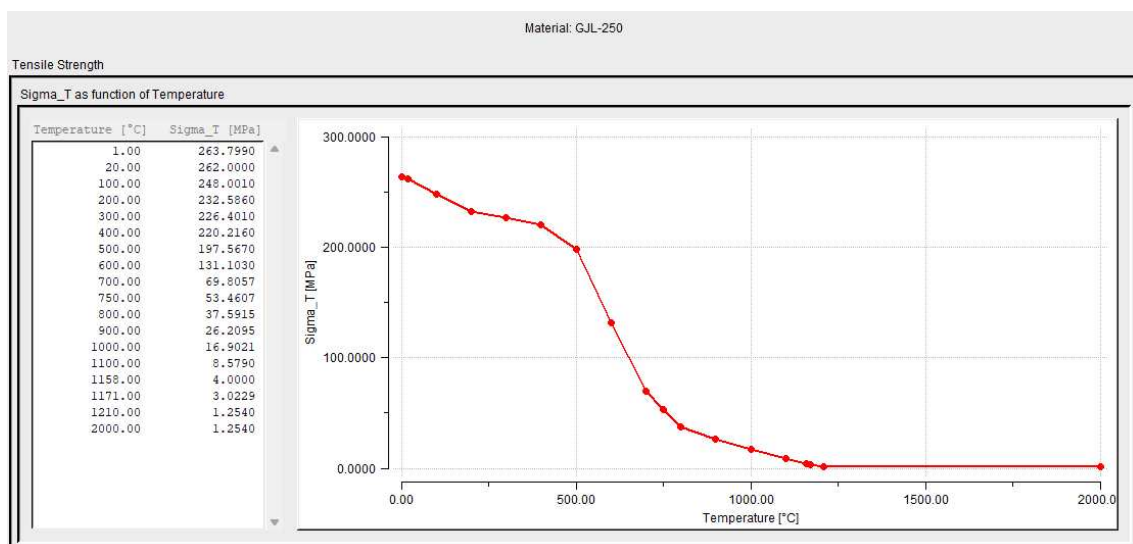
Feeding effectivity: %

Surface Tension Coefficient: N/m

Rheology model:

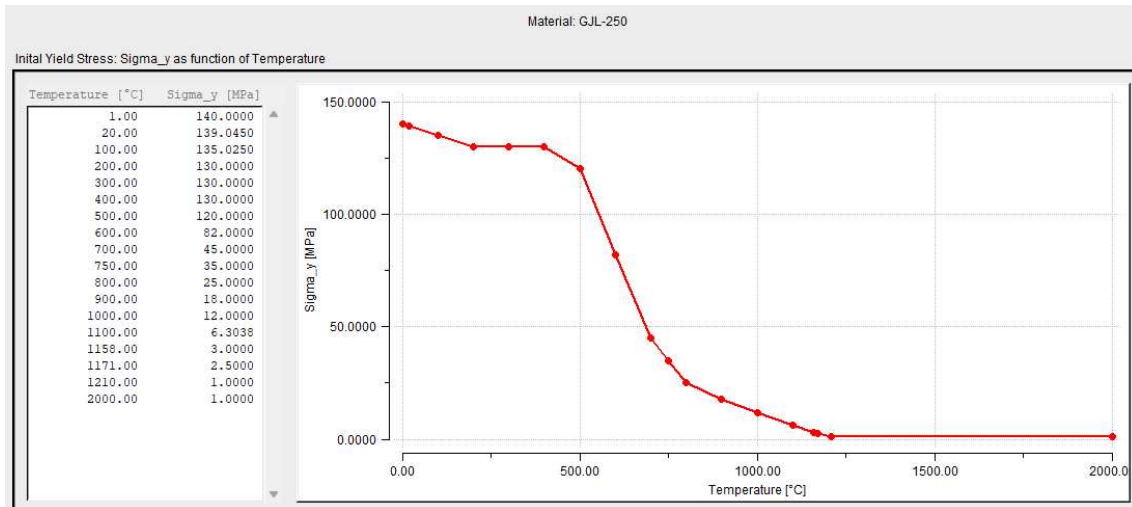
Fonte: Software Magmasoft

Curva da tensão de ruptura (σ_T) em função da temperatura do material utilizado
(GJL-250)



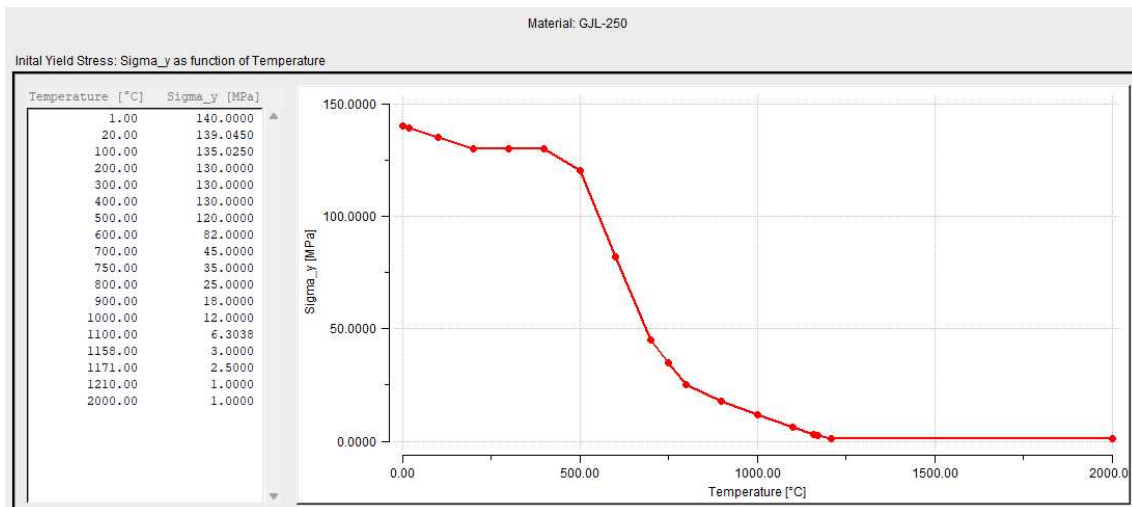
Fonte: Software Magmasoft

Curva da tensão de escoamento (σ_y) em função da temperatura do material utilizado (GJL-250)



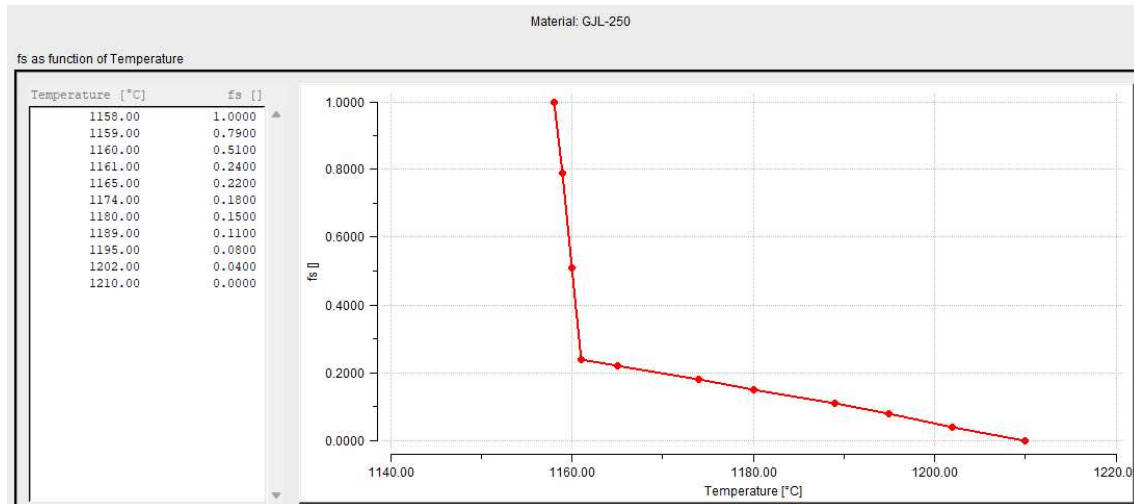
Fonte: Software Magmasoft

Curva do módulo de elasticidade (E) em função da temperatura do material utilizado (GJL-250)



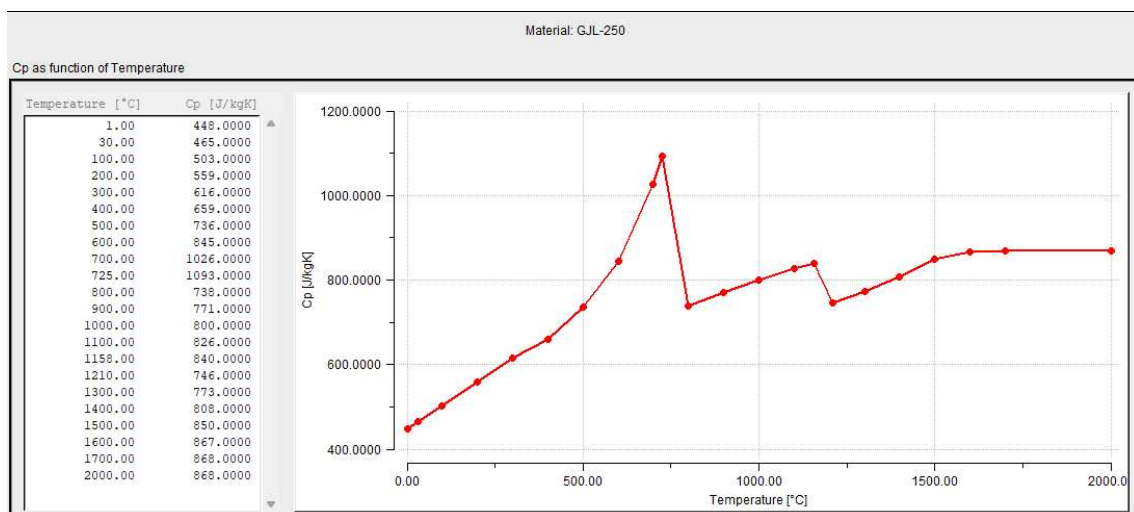
Fonte: Software Magmasoft

*Curva da fração de sólido (f_s) em função da temperatura do material utilizado
(GJL-250)*



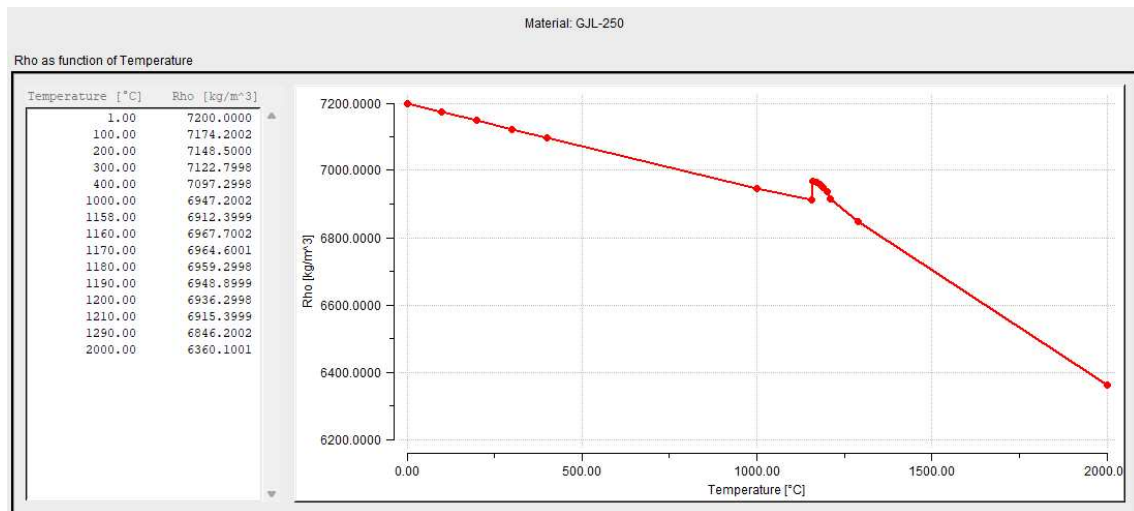
Fonte: Software Magmasoft

*Curva do calor específico (C_p) em função da temperatura do material utilizado
(GJL-250)*



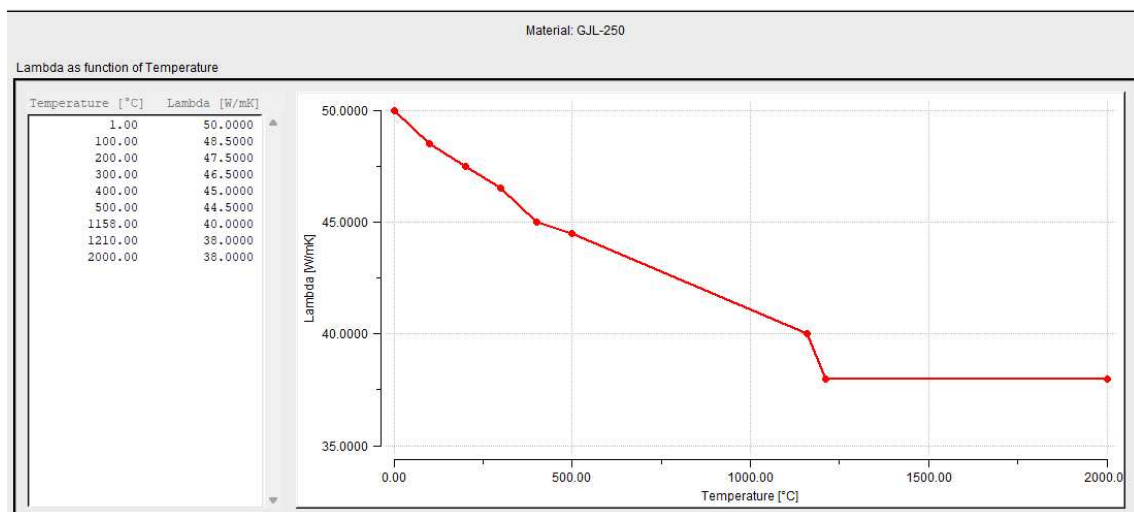
Fonte: Software Magmasoft

Curva da massa específica (ρ) em função da temperatura do material utilizado (GJL-250)



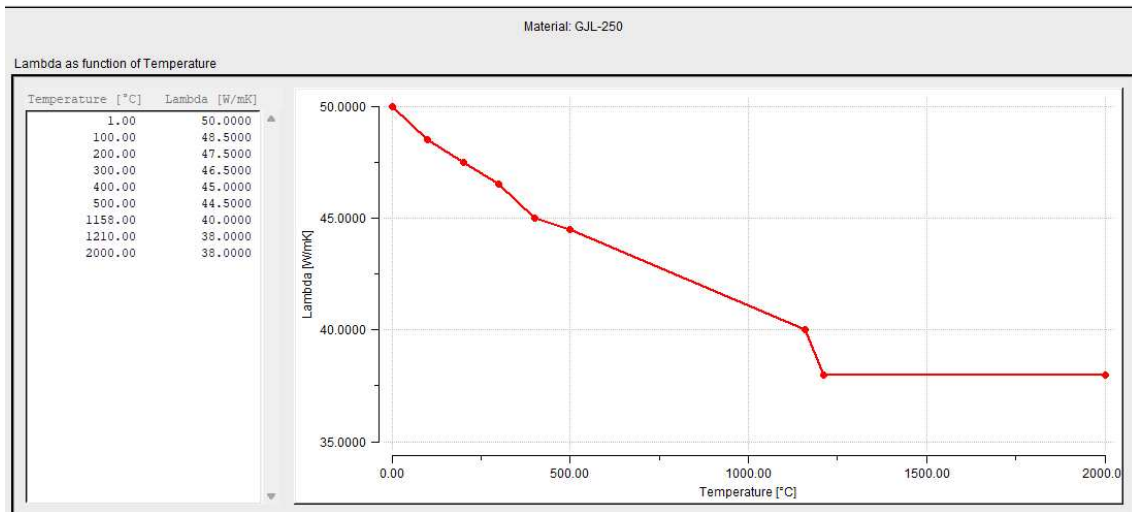
Fonte: Software Magmasoft

Curva do coeficiente de condutividade térmica (λ) em função da temperatura do material utilizado (GJL-250)



Fonte: Software Magmasoft

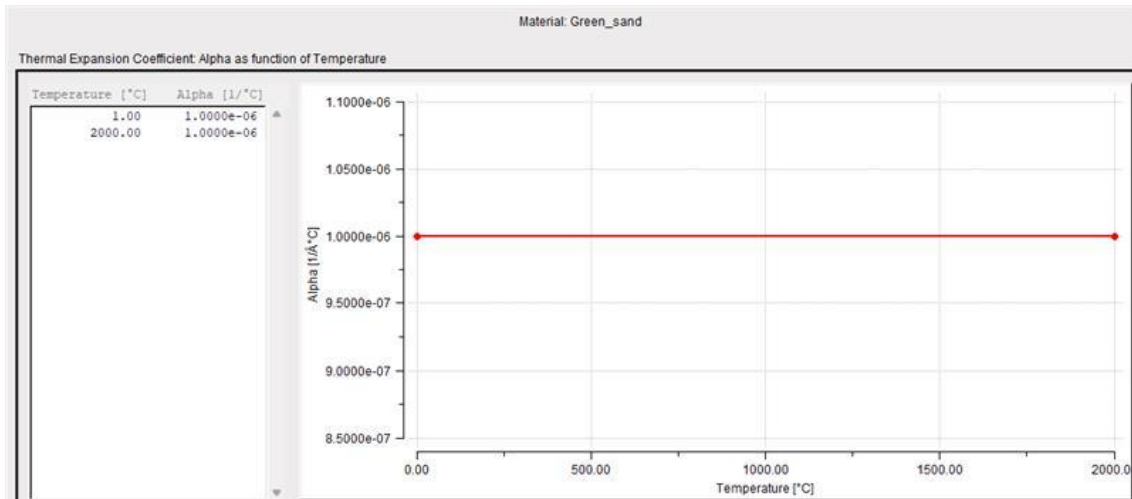
Curva do coeficiente de dilatação térmica (α) em função da temperatura do material utilizado (GJL-250)



Fonte: Software Magmasoft

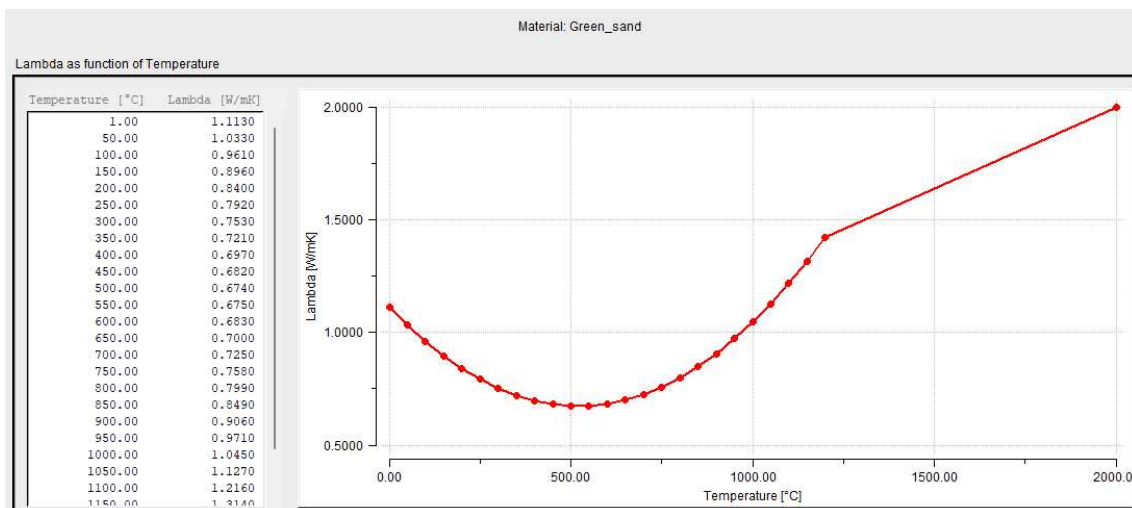
APÊNDICE B – PROPRIEDADES DA AREIA UTILIZADA

Curva do coeficiente de dilatação térmica (α) em função da temperatura da areia utilizada (areia verde)



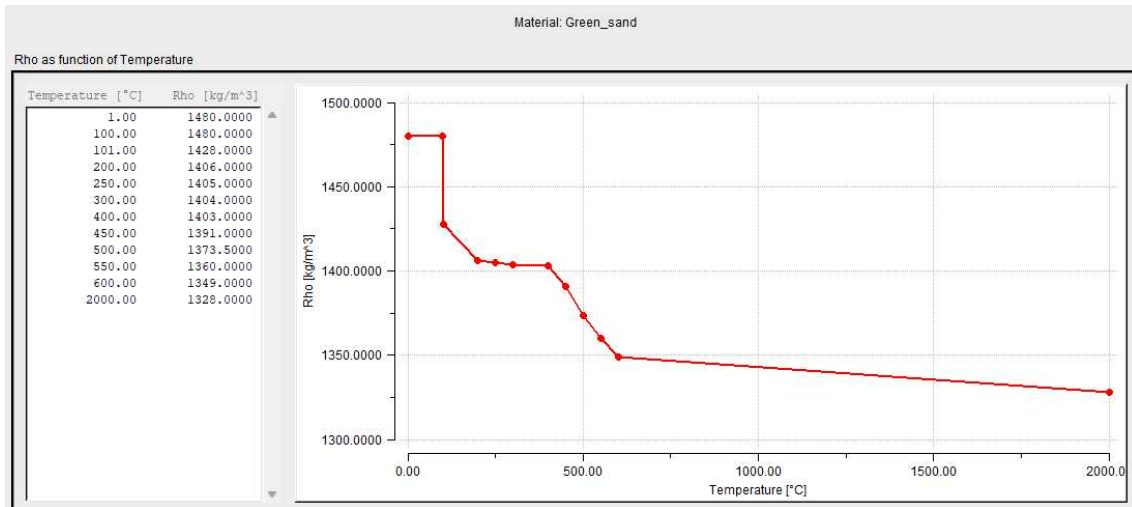
Fonte: Software Magmasoft

Curva do coeficiente de condutividade térmica (λ) em função da temperatura da areia utilizada (areia verde)



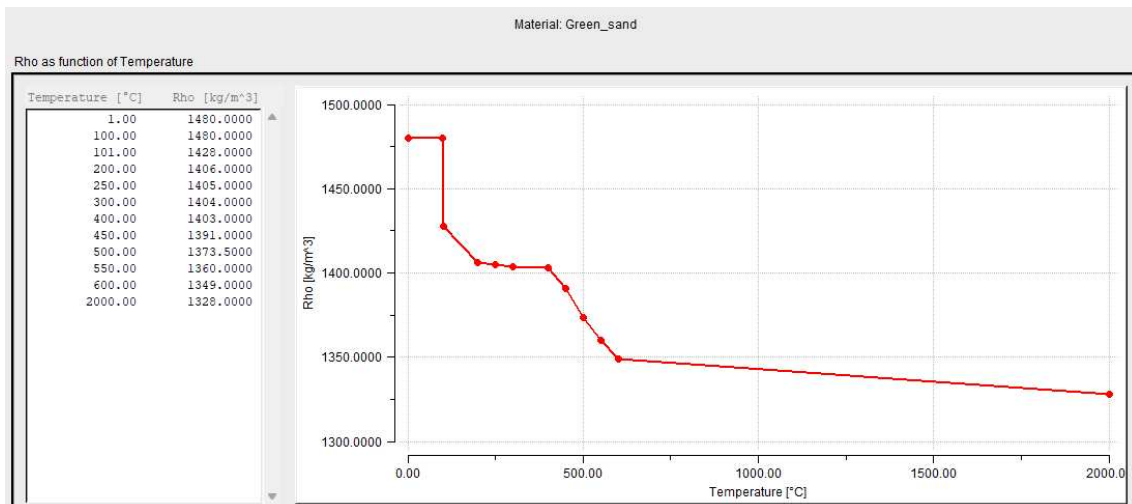
Fonte: Software Magmasoft

*Curva da massa específica (ρ) em função da temperatura da areia utilizada
(areia verde)*



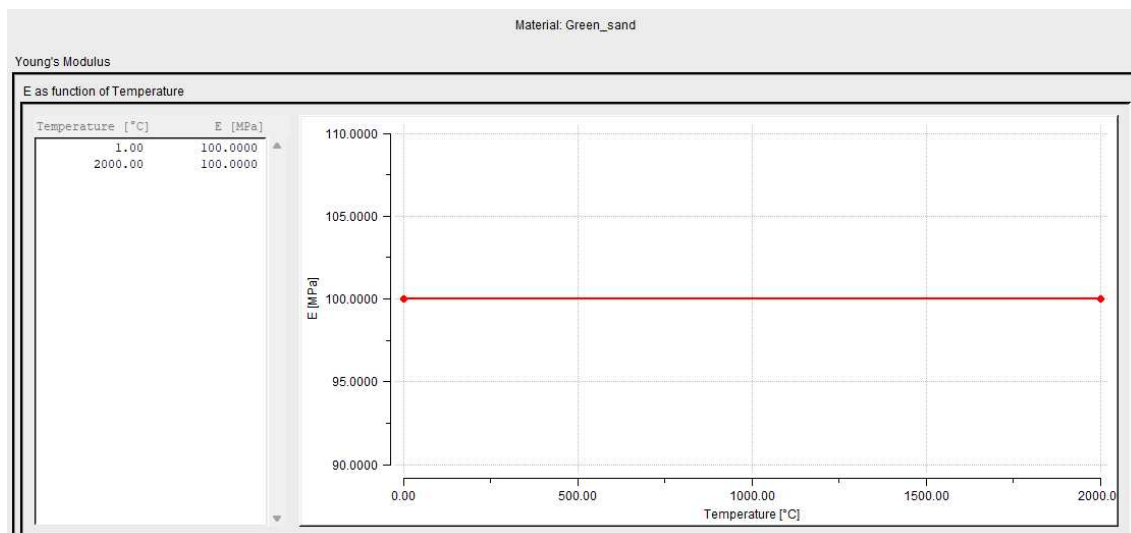
Fonte: Software Magmasoft

*Curva do calor específico (C_p) em função da temperatura da areia utilizada
(areia verde)*



Fonte: Software Magmasoft

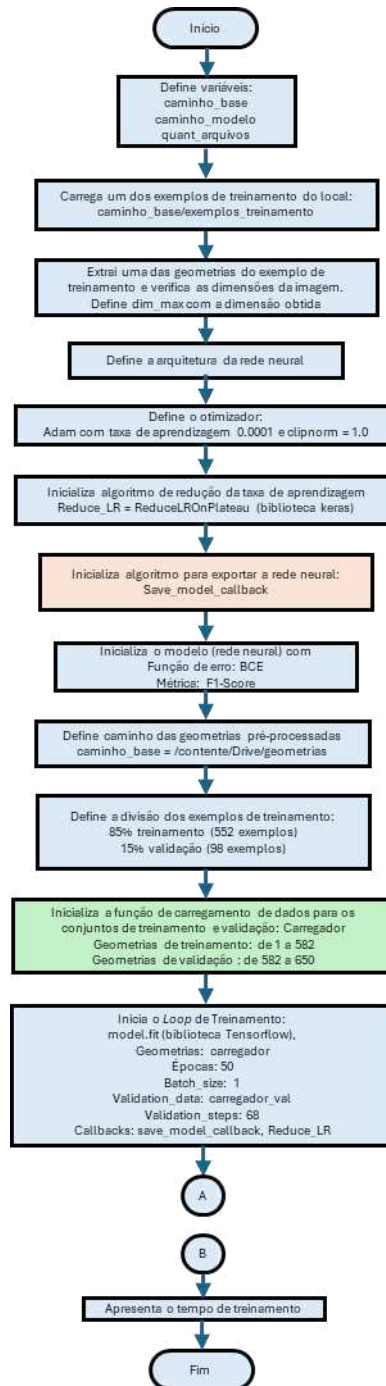
Curva do módulo de elasticidade (E) em função da temperatura da areia utilizada (areia verde)



Fonte: Software Magmasoft

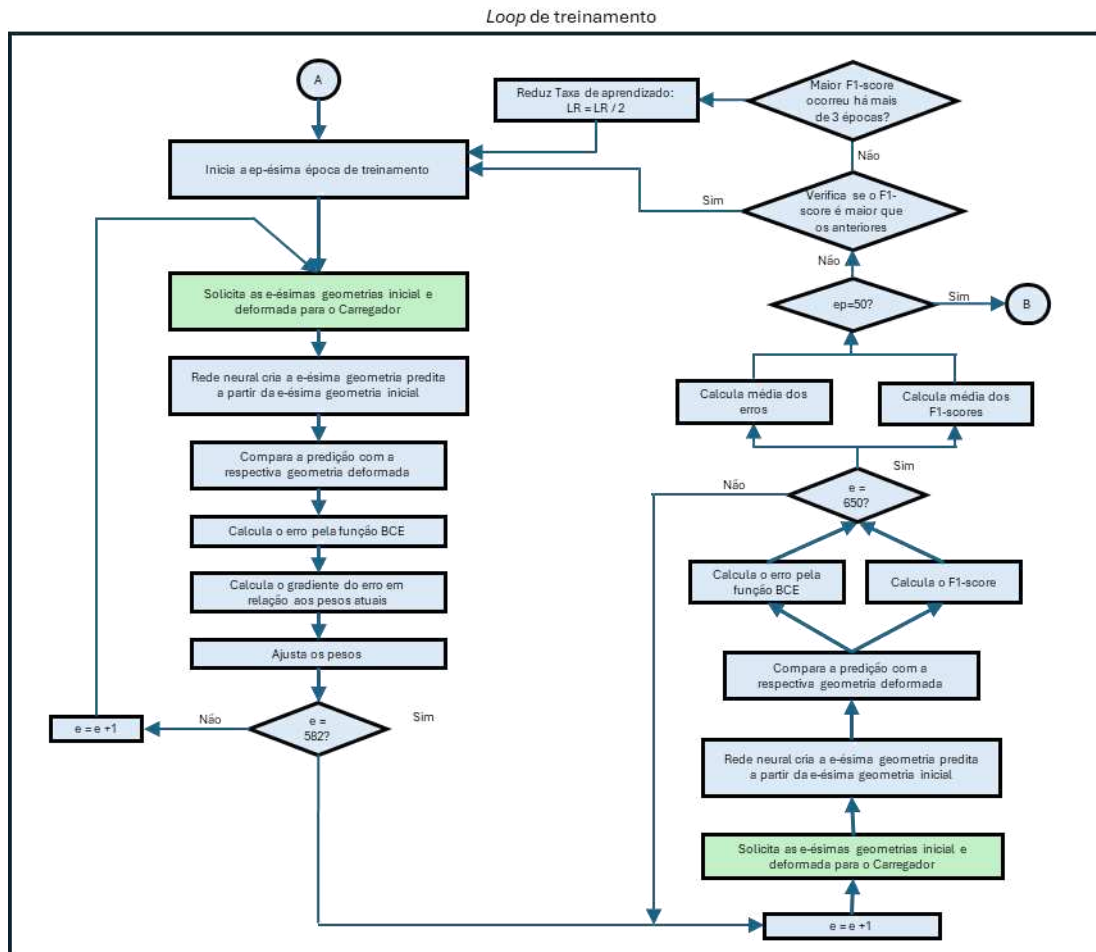
APÊNDICE C – FLUXOGRAMA DO PROGRAMA DE TREINAMENTO

Fluxograma do programa de treinamento da rede neural



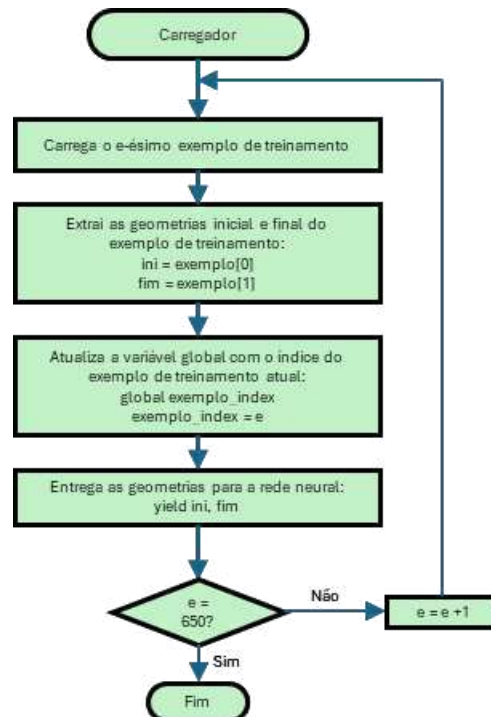
Fonte: o Autor

Fluxograma do loop de treinamento



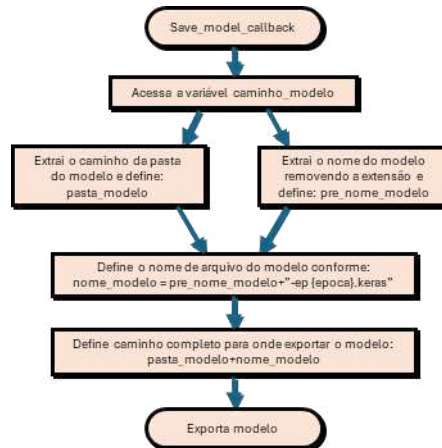
Fonte: o Autor

Fluxograma da função de carregamento das geometrias



Fonte: o Autor

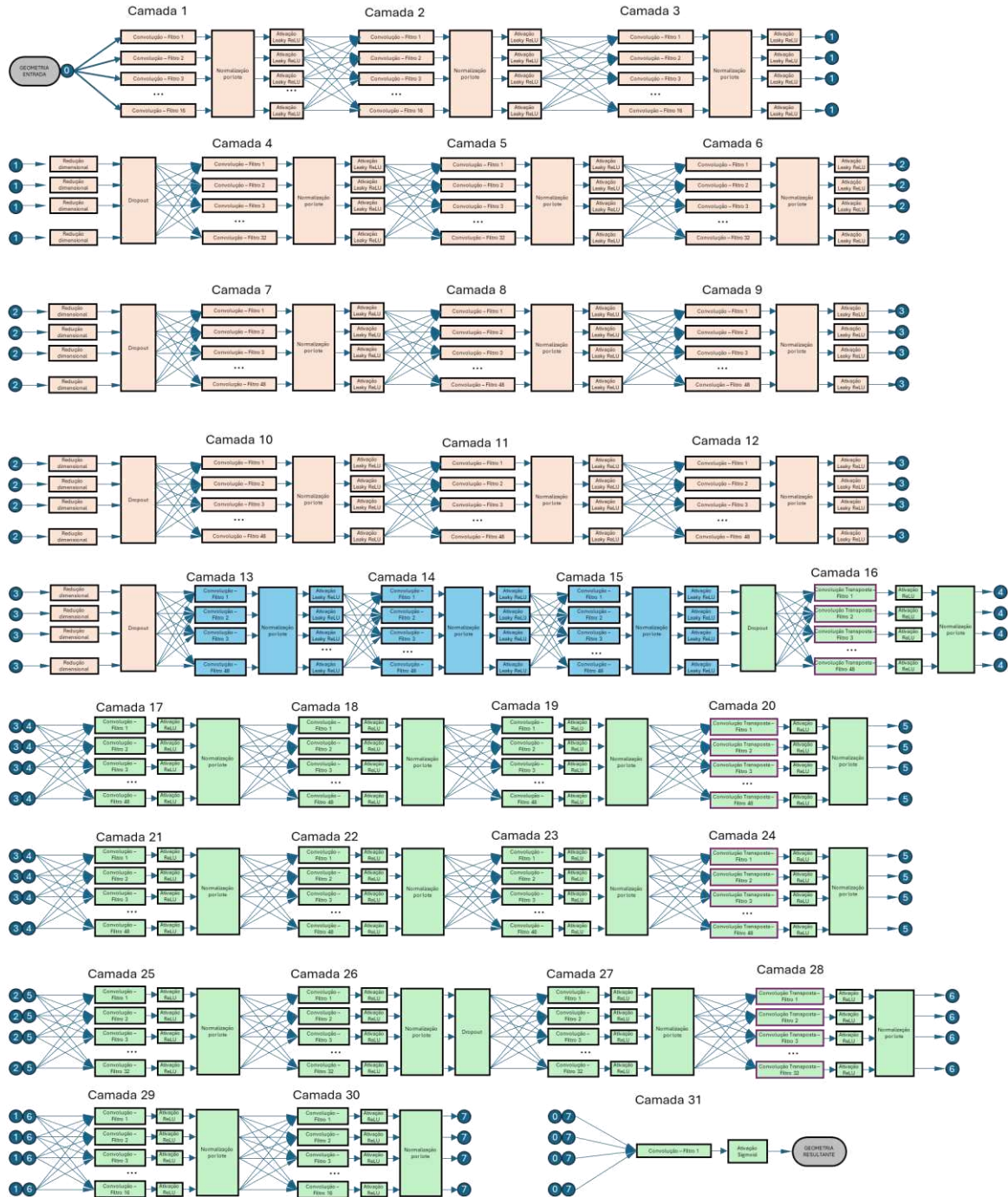
Fluxograma da função de exportação do modelo



Fonte: o Autor

APÊNDICE D – ARQUITETURA DA REDE NEURAL

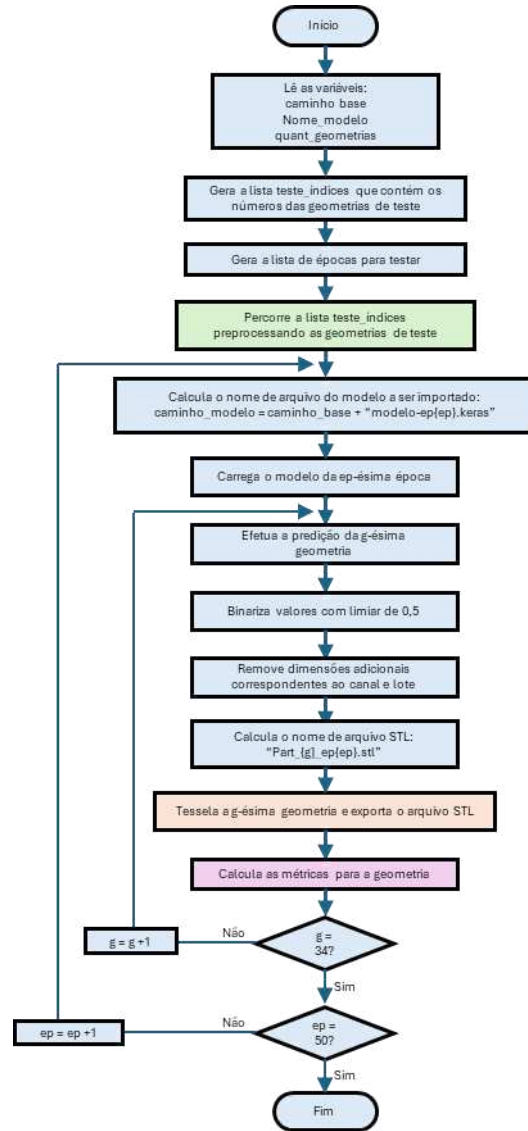
Arquitetura da rede neural



Fonte: o Autor

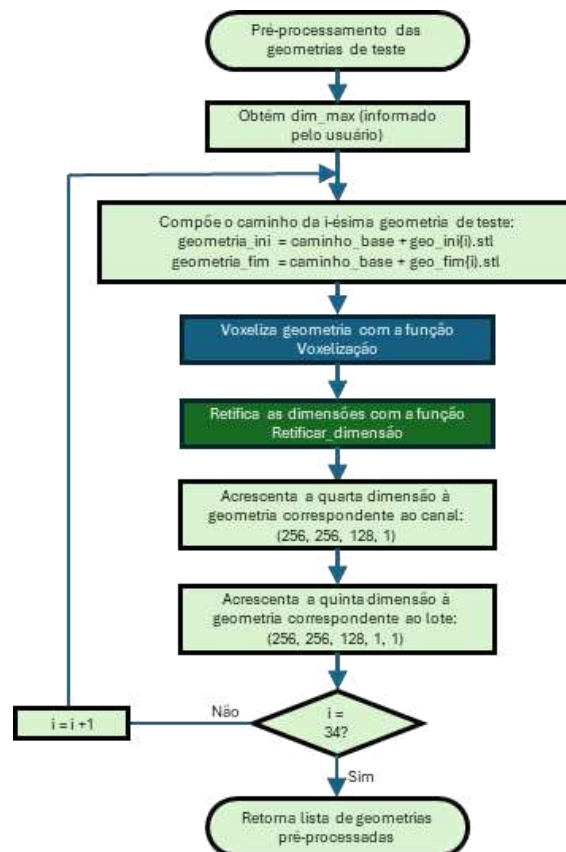
APÊNDICE E – FLUXOGRAMA DO PROGRAMA DE TESTES

Fluxograma do programa principal



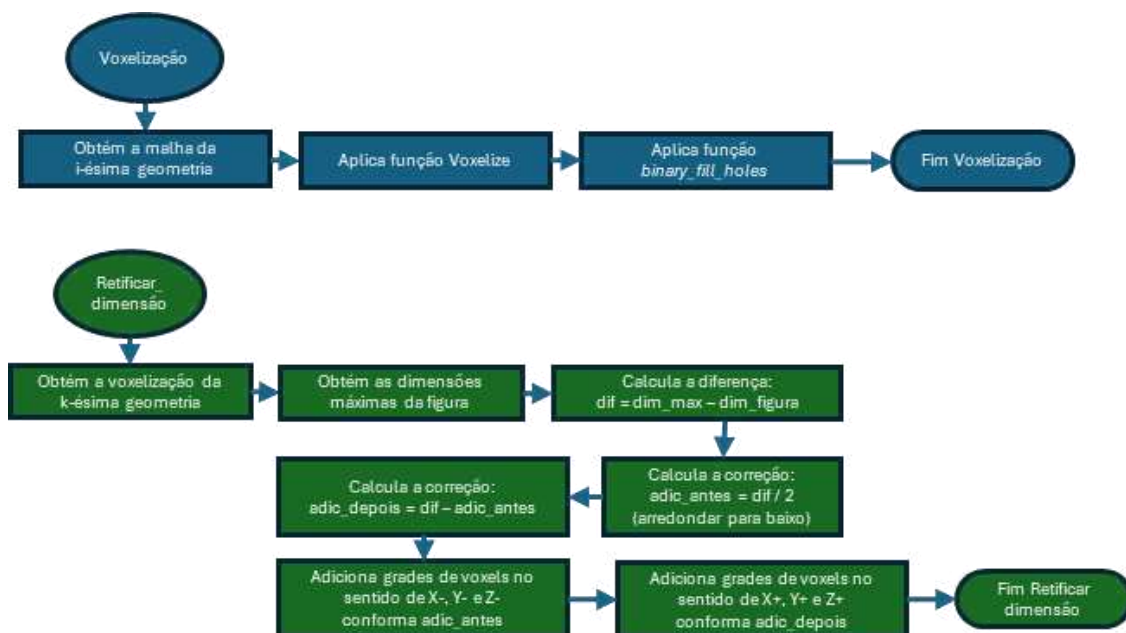
Fonte: o Autor

Fluxograma da função de pré-processamento das geometrias



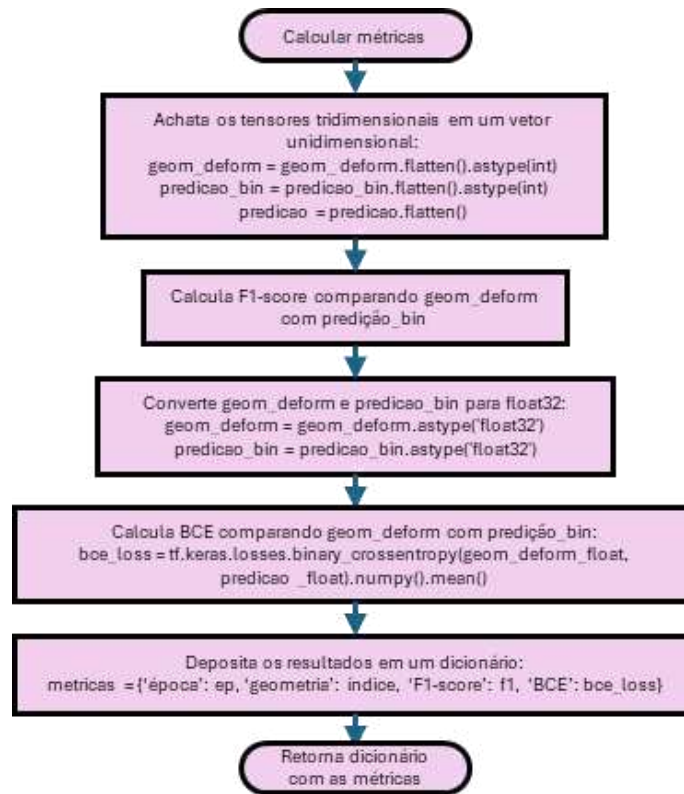
Fonte: o Autor

Fluxograma das funções de voxelização e retificar dimensão



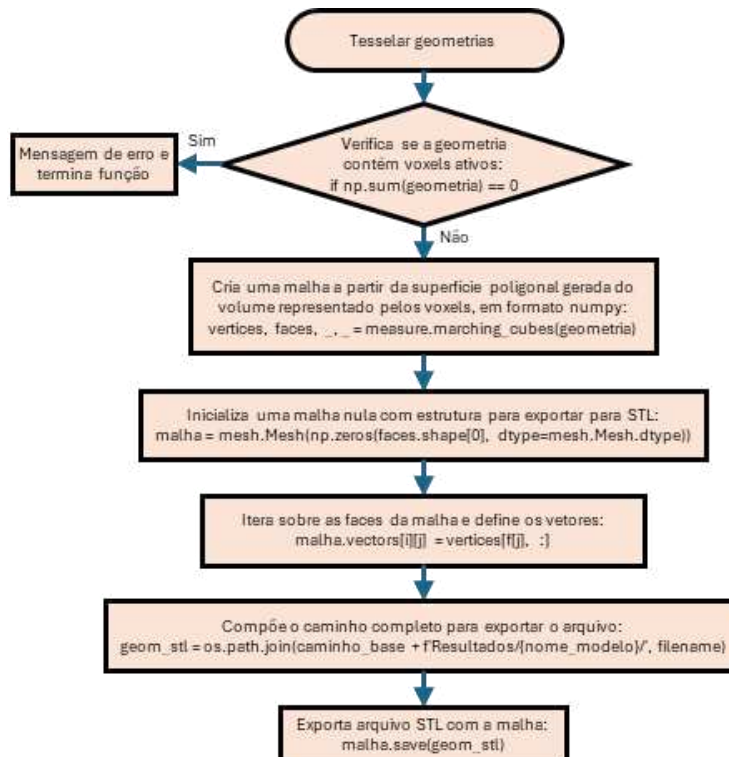
Fonte: o Autor

Fluxograma da função de cálculo das métricas



Fonte: o Autor

Fluxograma da função de cálculo das métricas



Fonte: o Autor