

**UNIVERSIDADE DE ARARAQUARA
PROGRAMA DE PÓS-GRADUAÇÃO EM BIOTECNOLOGIA EM MEDICINA
REGENERATIVA E QUÍMICA MEDICINAL**

JULIANO MARCELLO

**DESENVOLVIMENTO DE UM MÓDULO DE EXTENSÃO (PLUGIN) PARA O
SOFTWARE INVESALIUS VISANDO À INTEGRAÇÃO DE ETAPAS DA
BIOIMPRESSÃO 3D**

Araraquara, SP

2024

JULIANO MARCELLO

**DESENVOLVIMENTO DE UM MÓDULO DE EXTENSÃO (PLUGIN) PARA O
SOFTWARE INVESALIUS VISANDO À INTEGRAÇÃO DE ETAPAS DA
BIOIMPRESSÃO 3D**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Biotecnologia em Medicina Regenerativa e Química Medicinal da Universidade de Araraquara – UNIARA – como parte dos requisitos para obtenção do título de Doutor em Biotecnologia, Área de Concentração: Biotecnologia em Medicina Regenerativa e Química Medicinal.

Orientador:

Prof. Dr. Rodrigo Alvarenga Rezende

Coorientador:

Prof. Dr. André Capaldo Amaral

Araraquara, SP

2024

FICHA CATALOGRÁFICA

M263d Marcello, Juliano

Desenvolvimento de um módulo de extensão (PLUGIN) para o software Invesalius visando à integração de etapas da bioimpressão 3D/Juliano Marcello. – Araraquara: Universidade de Araraquara, 2024.

115f.

Tese (Doutorado) - Programa de Pós-Graduação em Biotecnologia em Medicina Regenerativa e Química medicinal – UNIARA

Orientador: Prof. Dr. Rodrigo Alvarenga Rezende

Coorientador: Prof. Dr. André Capaldo Amaral

1. Módulo de extensão. 2. Software InVesalius. 3. Biotecnologia. 4. Bioimpressão 3D. 5. Medicina regenerativa. 6. Tecnologia da Informação. I. Título.

CDU 577.4

JULIANO MARCELLO


Tese apresentada a Universidade de Araraquara - UNIARA, como parte dos requisitos para obtenção do título de Doutor em Biotecnologia em Medicina Regenerativa e Química Medicinal.

Araraquara, 01 de março de 2024.


BANCA EXAMINADORA

Prof. Dr. RODRIGO ALVARENGA REZENDE (Orientador)
Universidade de Araraquara – UNIARA

Prof. Dr. HERNANÉ DA SILVA BARUD
Universidade de Araraquara – UNIARA

 Documento assinado digitalmente
JOSE LUIS DAVILA SANCHEZ
Data: 04/03/2024 15:31:10 -0500
Verifique em <https://brasil.gov.br>

Dr. JOSE LUIS DÁVILA SÁNCHEZ
593iCAN Soluções em Impressão 3D Ltda

 Documento assinado digitalmente
ANA LUIZA GARCIA MILLAS MASSAGUER
Data: 07/03/2024 13:29:46 -0500
Verifique em <https://brasil.gov.br>

Dr^a ANA LUIZA GARCIA MILLAS MASSAGUER
Biotechnology Solutions (3DBS)

Prof. Dr. JEFFREY LEON-PULIDO
Universidad EAN – Colômbia

AGRADECIMENTOS

Aos meus amados pais, Suely Sedenho Marcello e Osmar Marcello. Seu apoio incondicional à minha carreira profissional e acadêmica é um presente inestimável que carrego com profundo carinho. Independentemente dos desafios, eles sempre estiveram ao meu lado, encorajando-me a buscar minha formação onde fosse necessário. Sou imensamente grato por sua dedicação e confiança em mim, valores que carrego em cada passo da minha trajetória.

À minha esposa e meu filho por todo o apoio incondicional que me deram durante o meu período acadêmico. Suas palavras de encorajamento e amor constante foram a força motriz por trás das minhas conquistas, e cada passo que dei em direção aos meus objetivos foi possível graças a você. Sou verdadeiramente abençoado por ter uma família tão incrível que sempre esteve ao meu lado.

À Universidade de Araraquara – Uniara pelo valioso papel desempenhado em minha jornada acadêmica. Aos colaboradores da instituição pelo profissionalismo e acolhimento nas dependências da instituição. Em especial gostaria de agradecer profundamente ao Reitor Prof. Dr. Luiz Felipe Cabral Mauro pela confiança em meu trabalho profissional e pelo apoio em meus estudos e crescimento acadêmico.

Às secretarias do Programa de Pós-Graduação em Biotecnologia em Medicina Regenerativa e Química Medicinal da Uniara, Sandra e Rosana pela amizade, pelo carinho, dedicação e ajuda durante minha permanência no programa, me auxiliando com a documentação necessária e procedimentos do programa.

À startup 3DBS e a seus proprietários Dra. Ana Luiza Millás e Dr. Pedro Massaguer pela atenção dispensada em reuniões e treinamentos que me ajudaram a entender o processo de bioimpressão 3D e pelo empréstimo de equipamentos para testes de impressão.

Ao CTI Renato Archer e aos desenvolvedores do software InVesalius Paulo Amorim e Thiago Moraes que dedicaram seu tempo para compartilhar seu conhecimento de valor inestimável para a conclusão do trabalho.

Ao meu orientador Dr. Rodrigo Alvarenga Rezende. Sua orientação atenciosa e experiência inestimável moldaram meu desenvolvimento acadêmico, permitindo-me enfrentar desafios complexos e avançar no campo da bioengenharia e da biofabricação. A colaboração e amizade compartilhada foram um privilégio,

impulsionando meu crescimento profissional e pessoal de maneiras inestimáveis. Agradeço profundamente por essa oportunidade única e pelo apoio essencial em minha jornada de pesquisa.

E agradeço ao Prof. Dr. André Capaldo Amaral que me incentivou a entrar no Programa de Pós-Graduação em Biotecnologia em Medicina Regenerativa e Química Medicinal da Uniara. Seus ensinamentos elucidaram meus pensamentos quanto a relação entre a tecnologia da informação e a biotecnologia e como a tecnologia da informação contribui para o desenvolvimento da biotecnologia.

RESUMO

O objetivo da Medicina Regenerativa e da Engenharia de Tecidos é regenerar ou substituir tecidos perdidos ou danificados, a fim de salvar ou oferecer melhores condições de vida aos pacientes. A bioimpressão 3D surge como uma tecnologia promissora para a fabricação de tecidos ou órgãos de forma automatizada amparada na aplicação de Tecnologias da Informação (TI). Cada tecido ou órgão possui características específicas e complexas - com estruturas morfológicas, composições celulares e moleculares próprias - além de funcionalidade bem definida. A fim de se reproduzir estruturas anatômicas é fundamental conhecer as especificidades de cada uma delas de modo a se projetar e integrar adequadamente as etapas da biofabricação, desde a criação da imagem computadorizada – *blueprint*, passando-se pela materialização por bioimpressoras até a fase de maturação tecidual feita por biorreatores. Nesse contexto, a extração da arquitetura do tecido/órgão e a geração de projetos de alta resolução são tarefas desafiadoras para a regeneração de tecidos. A proposta desse trabalho foi criar um módulo de extensão (*plugin*) ao *software InVesalius*, uma ferramenta que gera o modelo 3D digital construído a partir de imagens médicas obtidas por equipamentos, como por exemplo, de tomografia computadorizada ou ressonância magnética, para que seja possível segmentar o modelo 3D gerado e enviá-lo diretamente à bioimpressora 3D, evitando-se exportar o modelo para outros *softwares* de tratamento, antes de enviar para a bioimpressora, encurtando-se, assim, o caminho. Também foram desenvolvidos recursos para que os pesquisadores (usuários) possam registrar, na forma de arquivo, e gerenciar as informações relacionadas aos parâmetros operacionais de impressão, permitindo validar e armazenar as melhores práticas para impressões futuras, possibilitando otimizações no processo e potencializando os resultados. Em suma, o projeto facilita o processo de bioimpressão pelos usuários de qualquer formação tecnológica, principalmente aqueles sem amplo domínio e familiaridade com uso de *software* especializados.

Palavras-chave: Módulo de extensão; *Software InVesalius*; Biotecnologia; Bioimpressão 3D; Medicina Regenerativa; Tecnologia da Informação.

ABSTRACT

The objective of Regenerative Medicine and Tissue Engineering is to regenerate or replace lost or damaged tissues in order to save or offer better living conditions to patients. 3D bioprinting emerges as a promising technology for manufacturing tissues or organs in an automated manner supported by the application of Information Technologies (IT). Each tissue or organ has specific and complex characteristics - with its own morphological structures, cellular and molecular compositions - in addition to well-defined functionality. To reproduce anatomical structures, it is essential to know the specificities of each one of them to properly design and integrate the stages of biofabrication, from the creation of the computerized image – blueprint, through materialization by bioprinters to the tissue maturation phase made by bioreactors. In this context, the extraction of the tissue/organ architecture and the generation of high-resolution designs are challenging tasks for tissue regeneration. The purpose of this work was to create an extension module (plugin) for the InVesalius software, a tool that generates a digital 3D model built from medical images obtained by equipment, such as CT scan (computed tomography) or MRI (magnetic resonance imaging), so that it is possible to segment the generated 3D model and send it directly to the 3D bioprinter, avoiding exporting the model to other treatment software before sending it to the bioprinter, thus shortening the path. Resources were also developed so that researchers (users) can record, in file form, and to manage the information related to the printing operational parameters, allowing them to validate and store the best practices for future printing, enabling process optimization and enhancing results. In short, the project facilitates the bioprinting process for users of any technological background, especially those without extensive knowledge and familiarity with the use of specialized software.

Keywords: Extension module; InVesalius Software; Biotechnology; 3D bioprinting; Regenerative Medicine; Information Technology.

Lista de abreviaturas e siglas

2D	Bidimensional
3D	Tridimensional
3DBS	3D BIOTECHNOLOGY SOLUTIONS (<i>startup</i> brasileira)
4D	Tetradimensional
ACR	Colégio Americano de Radiologia
AMF	Arquivo de Fabricação Aditiva
API	Interface de Programação de Aplicação
ASCII	<i>American Standard Code for Information Interchange</i>
ASTM	<i>American Society for Testing and Materials</i>
C&T	Ciência e Tecnologia
CAD	<i>Computer-Aided Design</i> (Projeto Assistido por Computador)
CDA	<i>Clinical Document Architecture Release</i>
CTI	Centro de Tecnologia da Informação Renato Archer
DICOM	<i>Digital Imaging and Communications in Medicine</i>
DNA	Ácido Desoxirribonucleico
ET	Engenharia de Tecidos
FDA	<i>Food and Drug Administration</i>
FDM	<i>Fused Deposition Modeling</i>
GUI	<i>Graphical User Interface</i>
HL7	<i>Health Level 7</i>
HW	<i>Hardware</i>
ID	Identificador
ISO	Organização Internacional de Padronização
LCD	Display de Cristal Líquido
MA	Manufatura Aditiva
MicroSD	<i>Micro Secure Digital</i>
MIT	<i>Massachusetts Institute of Technology</i>
MR	Medicina Regenerativa
NEMA	Associação Nacional dos Fabricantes de Equipamentos Elétricos

Núcleo de Tecnologias Tridimensionais da Universidade de

NUT3D	Araraquara
OSI	<i>Open System Interconnection</i>
pH	Potencial Hidrogeniônico
POI	<i>Parameter Optimization Index</i>
RAM	<i>Random Access Memory</i>
SC	Sistema Computacional
SCM	<i>Source Code Management</i>
SLA	<i>Stereolithography</i> (Estereolitografia)
SO	Sistema Operacional
STL	<i>Standard Triangle Language</i>
SW	<i>Software</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TI	Tecnologia da Informação
USB	<i>Universal Serial Bus</i>
VTK	<i>Visualization Toolkit</i>
Wi-Fi	<i>Wireless Fidelity</i>
XML	<i>eXtensible Markup Language</i>

Sumário

1	Introdução.....	15
1.1	Medicina Regenerativa (MR) e Engenharia de Tecidos (ET)	17
1.2	A Manufatura Aditiva (MA) ou Impressão 3D	18
1.3	Biofabricação.....	20
1.3.1	A Biofabricação e suas estratégias.....	21
1.4	A Bioimpressão 3D.....	23
1.4.1	Desafios.....	24
1.5	Tecnologia da Informação na Bioimpressão 3D.....	25
1.5.1	DICOM - <i>Digital Imaging and Communications in Medicine</i>	29
1.5.2	Arquivo STL.....	31
1.5.3	Arquivo AMF	33
1.5.4	<i>InVesalius</i> : SW livre de imagens médicas	34
1.5.5	<i>Slicer</i> (Fatiamento)	35
1.5.6	<i>gCode</i>	37
2	Justificativa	40
3	Objetivos.....	44
3.1	Objetivos Específicos	44
4	Materiais e Métodos.....	45
4.1	Impressora e Bioimpressora 3D	45
4.2	Computador e sistema operacional de desenvolvimento	45
4.3	<i>InVesalius</i>	45
4.4	<i>Visual Studio Code</i>	46
4.5	Python	46
5	Desenvolvimento do Trabalho	49
5.1	Desenvolvimento da Interface do <i>BioSlicer</i>	55
5.2	<i>Features</i>	62

5.2.1	Visualização do modelo 3D	62
5.2.2	Parametrização.....	67
5.2.3	Algoritmo de Fatiamento (<i>Slicer</i>)	71
5.2.4	Código <i>gCode</i> gerado pelo algoritmo	72
5.2.5	Exportar/Importar <i>gCode</i>	75
5.2.6	Impressão do modelo	76
5.3	Validação do <i>plugin</i>	78
5.3.1	Teste de Impressão com Filamento.....	78
5.3.2	Teste de Impressão com Creme Hidratante e Hidrogel.....	84
5.3.3	Limitações atuais de uso	87
5.4	Ambiente de desenvolvimento do <i>software</i> proposto.....	87
5.4.1	Git.....	87
5.4.2	Chocolatey.....	88
5.4.3	Visual Studio 2019 <i>community edition</i>	89
5.4.4	<i>Python</i>	89
5.4.5	Compilando e executando o <i>InVesalius</i>	93
5.4.6	<i>wxFormBuilder</i>	94
5.4.7	Visual Studio Code	96
5.4.8	Pacotes adicionais do <i>Python</i>	98
6	Discussão e Trabalhos Futuros	99
6.1	Sugestões de trabalhos Futuros.....	101
7	Conclusão.....	103
8	Referências Bibliográficas	105
	APÊNDICE A – DEFINIÇÕES DO PADRÃO DICOM	112
	APÊNDICE B - INPI – DOCUMENTO DO REGISTRO DO SOFTWARE	115

Lista de Figuras

Figura 1: Processos de impressão 3D.	19
Figura 2: Inter-relação entre Biofabricação, Manufatura Aditiva, Engenharia de Tecidos (ET) e Medicina Regenerativa (MR) (Groll et al., 2016).....	21
Figura 3: Modelos de Bioimpressoras.	24
Figura 4: Relação entre as áreas de Biofabricação, Bioimpressão 3D e Tecnologia da Informação.	25
Figura 5: Interrelação da TI com os Sistemas Biológicos e a Engenharia.	26
Figura 6: Sequenciamento dos tipos de software usados no processo de bioimpressão.....	28
Figura 7: Modelo geral de comunicação DICOM.....	30
Figura 8: Estrutura Básica de um arquivo DICOM.	30
Figura 9: Codificação da superfície do objeto em triângulos.....	32
Figura 10: Imagem de ressonância magnética com o cérebro selecionado com a nova ferramenta de region growing.....	35
Figura 11: Abordagens no fatiamento.	36
Figura 12: Imagem fatiada no Ultimaker Cura.....	37
Figura 13: Exemplo de código escrito em gCode.....	39
Figura 14:Fases de um processo de biofabricação.....	50
Figura 15: Fluxograma das etapas da Bioimpressão por extrusão.	51
Figura 16: Etapas da bioimpressão encapsuladas pelo plugin.	53
Figura 17: Passos para acessar o plugin BioSlicer no InVesalius.....	54
Figura 18: Interface do Plugin BioSlicer.	55
Figura 19: Projeto da interface visualizado no wxFormBuilder.....	56
Figura 20: Código gerado pelo wxFormBuilder.	58
Figura 21: Primeira Versão da interface do Plugin.....	59
Figura 22: Versão atual da Interface do BioSlicer.	60
Figura 23: Frame - Configuração de Impressão.....	61
Figura 24: Frame - Configuração da Impressora.....	61
Figura 25: Frame - Imprimir.....	62

Figura 26: Barra de ferramentas do visualizador 3D.	64
Figura 27: Visualizador 3D com Cursor de Rotação Habilitado.....	64
Figura 28: Objeto 3D com a opção editar habilitada.	66
Figura 29: Função Ver/Ocultar Limites da Impressora.	67
Figura 30: Formato de arquivos STL em ASCII.....	72
Figura 31: gCode gerado no BioSlicer.	74
Figura 32: Opções de Exportar/Importar gCode.....	75
Figura 33: Frame Imprimir com a exibição do progresso da impressão.	77
Figura 34: Segmentação de imagem para testes de impressão.	79
Figura 35: Comparação entre o gCode gerado pelo Ultimaker e o BioSlicer.	80
Figura 36: Antes e depois de suavizar os pontos no gCode.	82
Figura 37: Comparação entre os modelos impressos no Ultimaker e no BioSlicer. ...	83
Figura 38: Modelo criado no Autodesk Inventor para teste na bioimpressora.....	85
Figura 39: Testes com Bioimpressora: à esquerda, impressão com hidrogel (alginato 2% e 7% gelatina); à direita impressão com creme hidratante.....	86
Figura 40: Versão do Chocolate.....	89
Figura 41: Janela Propriedades do Sistema do Windows 11.	90
Figura 42: Janela Editar variável de ambiente do usuário.....	91
Figura 43: Janela Editar variável de ambiente do sistema.	92
Figura 44: Instalação do wxFormBuilder Licenciamento.	94
Figura 45: Instalação do wxFormBuilder pasta de destino.	95
Figura 46: Instalação do wxFormBuilder - seleção dos componentes.	96
Figura 47: Instalação do Visual Studio Code - Licenciamento.	96
Figura 48: Instalação do Visual Studio Code – Local de Destino.	97
Figura 49: Instalação do Visual Studio Code – Tarefas a serem executadas.	97

1 Introdução

A humanidade tem testemunhado nas últimas décadas verdadeiras revoluções na Ciência e Tecnologia (C&T). Grandes áreas como as ciências biológicas e exatas, como a medicina, as engenharias, a computação e a biotecnologia, bem como diversas disciplinas, tais como a nanotecnologia, biomateriais, genética, bioquímica, robótica, inteligência artificial, *Big Data*, dentre várias outras, têm conjuntamente promovido um avanço como nunca visto.

A Biotecnologia possui relação direta e indireta com todas áreas e disciplinas destacadas acima. Sobretudo, o avanço da computação, a qual será tratada neste projeto pelo termo “Tecnologia da Informação (TI)”, tem impulsionado fortemente todos os setores de C&T.

A Biotecnologia é definida como qualquer técnica que utilize organismos vivos (ou partes de organismos), como plantas, animais ou microrganismos, engenheirados ou não, para produzir ou modificar produtos e processos, para usos específicos e com impacto na academia, na bioindústria e na bioeconomia (Capes, 2019). A Biotecnologia utiliza-se de ciências como estatística, matemática, engenharia e ciência da computação para interpretar e processar dados biológicos, automatizar processos, organizar dados, criar dispositivos etc.

Segundo a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) em seu documento de área de 2019, as pesquisas em Biotecnologia devem buscar estratégias em áreas da fronteira do conhecimento, tais como as, ciências ômicas, neurociências, células-tronco, biofármacos, conversão de biomassa e nanobiotecnologia, pois estas são promissoras em inovações tecnológicas de alto valor agregado.

O advento da tecnologia do DNA (Ácido Desoxirribonucleico) recombinante, na década de 1970, foi um marco na história da biotecnologia. A descoberta de enzimas de restrição permitiu aos cientistas cortar e recombinar segmentos de DNA de diferentes organismos, possibilitando a criação de organismos geneticamente modificados e a produção de medicamentos por meio da tecnologia de DNA recombinante (Brock, et al.,2003). Na década de 1990, dois cientistas da *Harvard Medical School*, Langer e Vacanti (1993),

realizaram um experimento que chamou a atenção não apenas da comunidade científica, mas da sociedade como um todo. Uma estrutura projetada, fabricada com células cartilaginosas de bovino, crescidas sobre uma matriz artificial no formato de uma orelha, foi implantada sob a pele no dorso de um camundongo, com o intuito de ser amadurecida. O camundongo ficou bastante famoso como o “rato-orelha”. A partir daquele momento, a medicina regenerativa (MR) e engenharia de tecidos (ET) passaram a estar sob holofote permanente da mídia e da academia. Esta notoriedade alavancou a área e promoveu uma verdadeira transformação.

Em paralelo a esses fatos, ocorria, entre os anos 1980 e 1990, o desenvolvimento a largos passos da indústria da Manufatura Aditiva (MA), sendo naquela altura ainda conhecida por “Prototipagem Rápida” e, atualmente, popularmente conhecida por “Impressão 3D”. A possibilidade real de se projetar estruturas com geometrias das mais simples às mais complexas e materializá-las por meio de um equipamento automatizado e de alta precisão e com um vasto leque de materiais, chamou a atenção de vários setores da indústria e academia, dentre os quais, o da MR e ET, que passou a adotar a tecnologia como uma importante ferramenta nos desenvolvimentos. Posteriormente, a ET beneficiou-se desta tecnologia originando a área de biofabricação de forma automatizada.

A MA é bastante utilizada para a biofabricação tridimensional de estruturas de suporte temporárias e biodegradáveis, conhecidas por *scaffolds* (termo em inglês, de andaimes ou arcabouços), construídos com biomateriais como polímeros (naturais e/ou sintéticos), cerâmicos, metais ou compósitos dos mesmos. Os *scaffolds* passaram então a ter células incorporadas com a proposta de promover a formação de um novo tecido humano com o objetivo de ajudar na regeneração de partes danificadas do corpo, causadas por doenças, acidentes ou doenças congênitas (Belo, 2024).

A Biofabricação é um campo de pesquisa incipiente e de rápido crescimento que continua a desenvolver inovações revolucionárias. Um progresso considerável também foi feito em tecnologias de MA, que são usadas para a produção de geometria complexa usando biomateriais, como andaimes e, mais recentemente, estruturas à base de biotinta (Montero *et al.*, 2019).

Com a evolução dos processos de MA, o que significa dizer o desenvolvimento de novas técnicas de fabricação, de biomateriais e de TI como

um todo, novas ideias surgiram, como a Bioimpressão de tecidos e órgãos humanos ou, simplesmente, Bioimpressão 3D.

1.1 Medicina Regenerativa (MR) e Engenharia de Tecidos (ET)

Há uma grande sobreposição entre os termos MR e ET. De fato, os objetivos são os mesmos, no entanto, a MR está voltada ao fim, e a ET ao meio, ou seja, de maneira geral, a ET é voltada ao desenvolvimento de técnicas e ferramentas, enquanto a MR as utiliza para atingir o objetivo final.

A MR, um campo interdisciplinar que aplica os princípios da engenharia e das ciências da vida para promover a regeneração, pode restaurar tecidos doentes e feridos e órgãos inteiros. Desde o início do campo há várias décadas, uma série de terapias de MR, incluindo aquelas projetadas para aplicações de cura de feridas e ortopedia, receberam a aprovação da *Food and Drug Administration* (FDA) e agora estão disponíveis comercialmente (Mao e Mooney, 2015).

A MR tem o potencial de curar ou substituir tecidos e órgãos danificados pela idade, doença ou trauma, bem como normalizar defeitos congênitos (Teixeira, 2024).

A ET é um campo interdisciplinar que aplica os princípios da engenharia e das ciências biológicas ao desenvolvimento de substitutos biológicos que restauram, mantêm ou melhoram a função do tecido (Langer e Vacanti, 1993).

O termo 'ET' foi usado pela primeira vez em meados da década de 1980 para descrever técnicas que eram usadas na manipulação de tecidos e órgãos em cirurgia. Em linhas gerais, o termo ET também foi referido às aplicações que envolviam o uso de biomateriais e dispositivos protéticos (Skalak e Fox, 1988).

Em 1987, o conceito de ET foi formalmente introduzido na medicina e definido como "ET é a aplicação dos princípios e métodos da engenharia e das ciências da vida para a compreensão fundamental das relações estrutura-função no tecido normal e patológico de mamíferos e o desenvolvimento de substitutos biológicos para restaurar, manter ou melhorar a função" (Lanza *et al.*, 2007).

Embora a colocação de células dentro de *scaffolds* não seja em algumas vezes bem controlada, devido à dificuldade de garantir a distribuição homogênea

no interior da estrutura, a bioimpressão 3D pode criar estruturas que combinam controle de alta resolução sobre o material e a colocação de células dentro de construtos projetados (Ozbolat, 2015).

A bioimpressão pode ser definida como a padronização espacial de células vivas e outros produtos biológicos, empilhando-os e montando-os usando uma abordagem de deposição camada por camada auxiliada por computador para desenvolver tecidos vivos e análogos de órgãos para ET, MR, farmacocinética e outros produtos biológicos estudados (Guillemot *et al.*, 2010).

1.2 A Manufatura Aditiva (MA) ou Impressão 3D

A MA, ou popularmente, impressão 3D é um processo de fabricação por meio da adição sucessiva de material na forma de camadas, com informações obtidas diretamente de uma representação geométrica computacional (Volpato, 2017). Desde o início da década dos anos 80 já havia o desenvolvimento do conceito de MA em nível de pesquisa, porém em 1986 o primeiro processo comercial foi lançado pela empresa americana *3D Systems*, denominado por SLA (*Stereolithography*), o qual é baseado no uso de laser ultravioleta para a cura de resinas fotossensíveis (Veit, 2018).

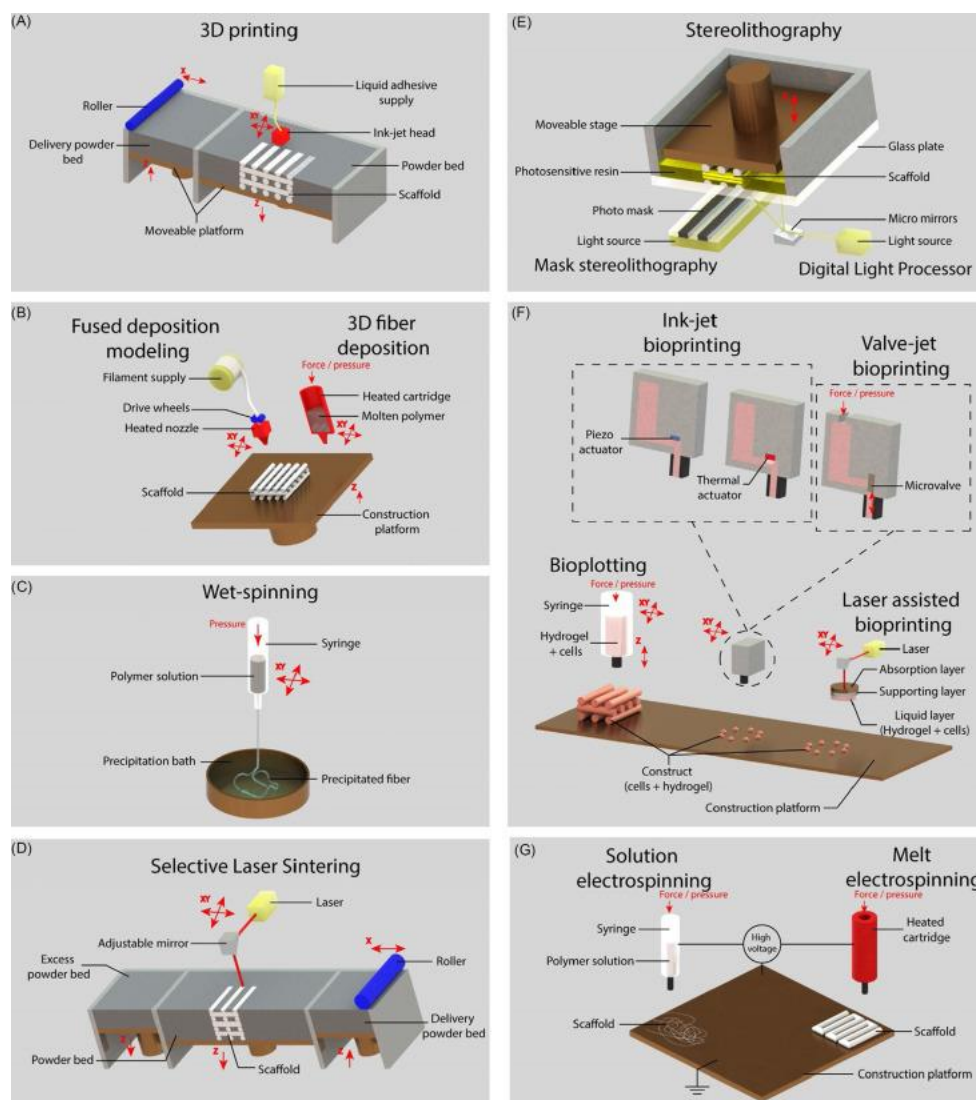
São mais de 30 processos comerciais de MA com importância econômica (Figura 1). Em 2012, o Comitê Internacional F42 de Fabricação de Aditivos da ASTM votou em uma lista de categorias, nomes e definições de processos (Silva e Rezende, 2013). Esses processos têm em comum o fato de produzirem objetos físicos a partir de uma representação de modelo virtual em um computador, frequentemente um modelo CAD - *Computer-Aided Design*, e estruturando continuamente uma camada de material sobre a outra. Isso abre novas possibilidades para produzir geometrias complexas ou impossíveis ao usar processos de produção regulares.

O processo de impressão 3D geralmente começa com a criação de um *blueprint* digital, que é chamado de arquivo *Computer-Aided Design* (CAD), que pode ser criado a partir do zero, obtido a partir de plataformas de serviços *online* especiais (*Thingiverse*, por exemplo) ou por digitalização 3D de um objeto desejado (Rayna e Striukova, 2016). O formato do arquivo CAD deve então ser

convertido para o arquivo *STereoLithography* (arquivo “STL” ou “.stl”) ou um Formato de Arquivo de Fabricação Aditiva (AMF) para ser interpretado por um *software* de segmentação (Gibson, 2015).

No seguinte passo, o objeto digitalmente projetado ou digitalizado é dividido em várias camadas e podem-se ajustar parâmetros específicos relativos, por exemplo, ao número de camadas ou sua espessura, a porcentagem de preenchimento, a temperatura do filamento, extremidade quente e leito, ou velocidade de impressão. A próxima etapa inclui a conversão desse arquivo em um *gCode*, o que permitirá que uma impressora 3D o leia, interprete e imprima o objeto desejado (Gibson, 2015).

Figura 1: Processos de impressão 3D.



Fonte: Kenny, 2019.

1.3 Biofabricação

A biofabricação é definida como a construção automatizada de produtos biologicamente funcionais, com organização estrutural de células vivas, biomoléculas, agregados celulares como esferoides e organoides utilizando várias estratégias, sendo uma delas a bioimpressão 3D (Groll *et al.*, 2016).

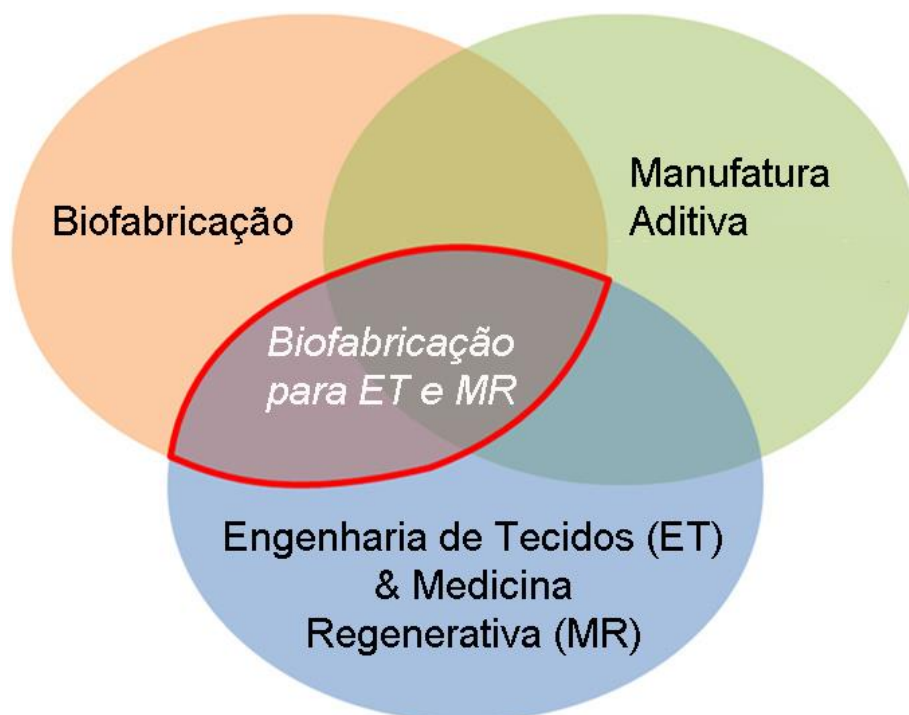
ET foi definida em 1993 como "*um campo interdisciplinar que aplica os princípios da engenharia e ciências biológicas para o desenvolvimento de substitutos biológicos que restauram, mantêm ou melhoram a função biológica do tecido ou de um órgão inteiro*" (Langer e Vacanti, 1993).

Em 2017, a definição de ET foi atualizada para "*o uso de processos físicos, químicos, biológicos e de engenharia para controlar e dirigir o comportamento agregado das células*" (Us Nstc, 2007).

O campo de MR foi definido como "*a aplicação da ciência dos tecidos, ET e princípios biológicos e de engenharia relacionados que restauram a estrutura e a função de tecidos e órgãos danificados*" (Us Dhhs, 2006).

Dentro da ET e MR, a Biofabricação potencializa soluções apoiando a ET e oferecendo alternativas para a MR (Figura 2).

Figura 2: Inter-relação entre Biofabricação, Manufatura Aditiva, Engenharia de Tecidos (ET) e Medicina Regenerativa (MR) (Groll et al., 2016).



Fonte: Adaptação de (Groll et al., 2016)

A biofabricação dentro da ET e MR visa explorar processos, em sua maioria técnicas de MA, para gerar construções de células-biomateriais que, por meio de seu arranjo espacial interno e externo, podem amadurecer em equivalentes de tecido funcional (Groll et al., 2016).

1.3.1 A Biofabricação e suas estratégias

De acordo com Dernowsek et al. (2015), as três estratégias da biofabricação são descritas como a seguir:

1.3.1.1 Estratégia baseada em Scaffolds:

A ET por Biomanufatura com o uso da MA partiu inicialmente de uma abordagem focada no uso de andaimes ou *scaffolds*, do inglês, que são estruturas tridimensionais biodegradáveis e biocompatíveis que têm um papel fundamental de oferecer suporte para adesão e proliferação de células abrindo

espaço para a formação de um novo tecido. As condições para o uso de *scaffolds* são:

- O *scaffold* deve garantir condições controladas quanto às propriedades mecânicas, químicas e topológicas adequadas para o projeto de regeneração de tecidos.
- Ao final, quando o novo tecido tiver sido formado, o *scaffold* deve ter sido totalmente degradado ou absorvido pelo organismo humano. Muitos biomateriais podem ser utilizados para a constituição de um *scaffold*.
- Os polímeros (naturais e sintéticos) são predominantes, mas os cerâmicos e metais podem também ser empregados, bem como, combinações entre todos esses.

A MA é excelente ferramenta para a construção de estruturas 3D a partir de geometrias complexas com alta precisão e tamanho de poro e porosidade controlados.

1.3.1.2 *Estratégia com base em biotintas (Bioimpressão 3D):*

A segunda estratégia é caracterizada pela aplicação de esferoides teciduais como blocos de construção para estruturas tridimensionais vivas.

Uma das primeiras definições de esferoides teciduais como blocos de construção para a bioimpressão foi dada por Mironov *et al.* (2009-a), em que microtecidos ou esferoides teciduais são materiais vivos com certa composição, material e propriedades biológicas mensuráveis, em evolução e potencialmente controláveis.

Mais recentemente, Moroni *et al.* (2018) atualizaram a definição dos esferoides teciduais - ou chamados pelos autores por esferoides celulares - como sendo um agrupamento de células com formato esférico, normalmente formado ao permitir que uma suspensão de células se fixe em uma gota de meio.

Durante o processo de bioimpressão, uma solução de um biomaterial ou uma mistura de vários biomateriais na forma de hidrogel, geralmente encapsulando os tipos de células desejados, denominada biotinta, é usada para criar construções de tecido (Gungor-Ozkerim *et al.*, 2018).

Segundo Groll *et al.* (2018), a biotinta é uma formulação de células adequadas para processamento por uma tecnologia de biofabricação automatizada que também pode conter componentes biologicamente ativos e biomateriais. Esta segunda abordagem demonstra que os tecidos biológicos podem ser projetados com composições e formas específicas, explorando a adesão característica célula-célula e a capacidade de cultura de células para desenvolver sua própria matriz extracelular.

1.3.1.3 *Estratégia híbrida baseada na associação das estratégias 1 e 2*

A terceira estratégia da biofabricação foi criada a partir da conexão entre as duas primeiras estratégias de forma a aproveitar as principais vantagens de cada uma delas. Por exemplo, o aproveitamento do reforço mecânico que um *scaffold* pode oferecer associado à característica de alta densidade celular dos esferoides teciduais. Um dos desenvolvimentos que exemplifica esta abordagem são as *lockyballs* (Rezende *et al.*, 2012).

1.4 A Bioimpressão 3D

Importantes descobertas relacionadas a biomateriais e biotecnologias tem ajudado a consolidar a bioimpressão 3D como um novo campo de aplicação da biomedicina e da ET (Silva e Duailibi, 2008).

A bioimpressão 3D é voltada à fabricação de estruturas 3D vivas com o objetivo de substituição de tecidos ou órgãos danificados ou comprometidos. O maior objetivo a longo prazo da bioimpressão é a fabricação de órgãos humanos funcionais para atender aqueles pacientes que se encontram em longas filas de espera por um órgão doado e compatível.

Os órgãos serão fabricados por bioimpressoras e terão como material principal células do próprio paciente. Esta condição diminuirá o risco de rejeição pelo organismo e reduzirá o uso de imunossupressores.

Figura 3: Modelos de Bioimpressoras.



Fonte: Composição feita pelo autor com imagens da Internet.

1.4.1 Desafios

Para se atingir esse objetivo de longo prazo, muitos desafios precisarão ser superados, de ordem tecnológica (biológicas, de materiais e de TI), de legislação e regulamentação, de ética, além de outros.

Dentre os obstáculos relacionados a TI, um dos grandes desafios é o desenvolvimento do chamado *blueprint* ou projeto da estrutura (anatomia) que se deseja produzir. O *blueprint* deve conter detalhes relacionados às características de material e de equipamentos, por exemplo.

Ainda com relação à TI, outro obstáculo é criar um sistema que integre os vários *softwares* (SWs) utilizados hoje nas várias etapas desde a aquisição e manipulação de imagens médicas, passando por SW de design até o SW de operação da bioimpressora.

Segundo uma das primeiras definições publicadas, o "projeto do órgão", especialmente no formato STL, é basicamente um programa de computador baseado em *software* que fornece instruções detalhadas para a colocação camada por camada de biocomponentes específicos usando um dispositivo

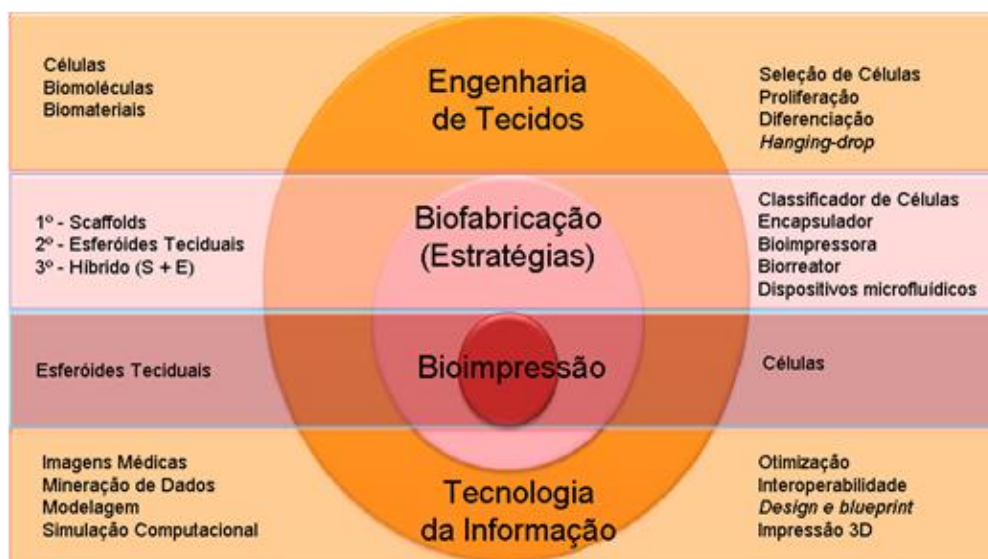
dispensador de acordo com o desenho assistido por computador - (CAD) original (Mironov *et al.*, 2008).

1.5 Tecnologia da Informação na Bioimpressão 3D

Segundo o relatório Gartner, Tecnologia da Informação é uma miríade de tecnologias de processamento de informação, incluindo *software*, *hardware*, tecnologias de comunicação e serviços relacionados, não incluindo tecnologias embarcadas que não geram dados para uso em geral. Essa mesma empresa define explicitamente a impressão 3D e a bioimpressão como ramos da TI (Gartner, 2015; Gartner IT, 2020).

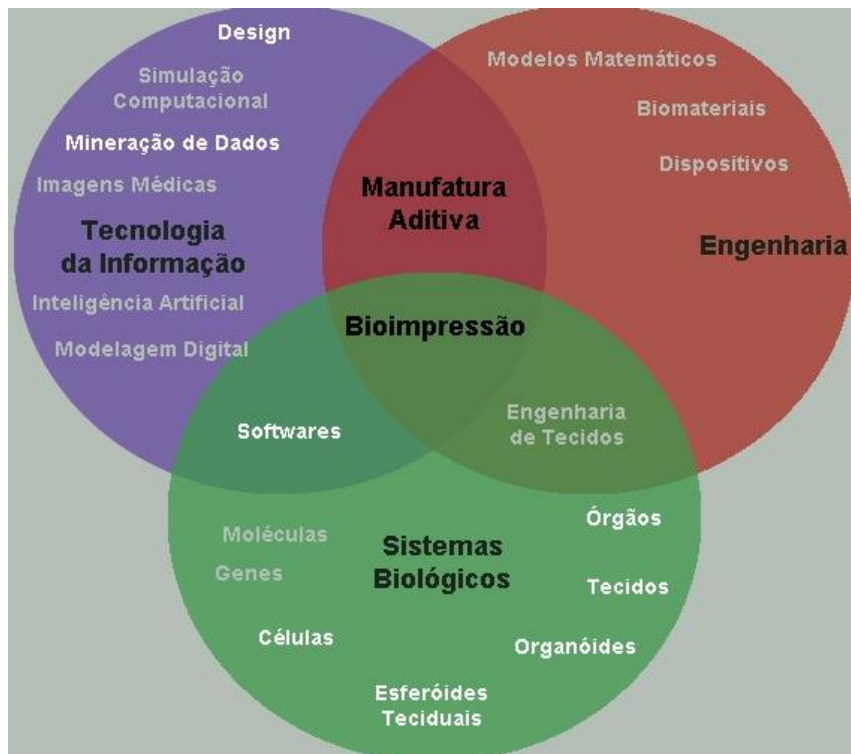
Em termos de bioimpressão de órgãos, o principal desafio para a tecnologia da informação de aplicação com o uso de biotintas com esferóides teciduais como blocos de construção é a ausência de *software* especializados que permitirá um projeto adequado e coerente de *blueprints* para a bioimpressão, embora haja certas tentativas de desenvolver tal *software* usando a abordagem de representação de função (Malone e Lipson, 2007; Rezende *et al.*, 2015). A Figura 4 mostra a relação entre Tecnologia da Informação e a Bioimpressão, enquanto a Figura 5 mostra a relação da Tecnologia da Informação com os diversos níveis biológicos estruturantes de um tecido.

Figura 4: Relação entre as áreas de Biofabricação, Bioimpressão 3D e Tecnologia da Informação.



Fonte: Dernowsek *et al.*, 2017 (adaptada pelo autor).

Figura 5: Interrelação da TI com os Sistemas Biológicos e a Engenharia.



Fonte: Produzida pelo autor.

A interoperabilidade dos sistemas de informação é baseada em padrões internacionais abertos, capazes de transmitir e interpretar dados e informações automaticamente. A representação atual de MA não é suficiente para a representação precisa de dados e informações para produtos e processos de biofabricação (Dernowsek *et al.*, 2017-a).

A descentralização no uso dos vários SW existentes que integram o fluxo de bioimpressão 3D dificulta o uso e o acesso a melhores resultados. Tanto a falta de ferramentas diretas para o projeto do *blueprint*, como da integração entre as existentes, limitam a biofabricação de estruturas complexas com representação de detalhes micro e macro anatômicos. A complexidade das construções deve ser aumentada e as células e componentes bioativos devem ser integrados (Visser *et al.*, 2013).

É crucial desenvolver novos padrões para dados e informações de biofabricação. Um futuro padrão de interoperabilidade de biofabricação incluirá dados e informações não apenas sobre micro e macro geometrias dos tecidos e

órgãos, mas também a posição correta de cada material, o destino celular e as biomoléculas necessárias, além de seus parâmetros de processamento (Dernowsek *et al.*, 2017-a).

Embora seja possível obter criar modelos digitais de árvores vasculares organoespecíficas, seja a partir de imagens médicas ou de programas CAD, e transformar em arquivos STL, esses 'projetos' não podiam ser usados diretamente de forma não modificada para *bioprinting* de órgãos moles, como fígado e rim, porque as construções de tecido vivo montadas a partir de esferoides de tecido vascular sofrem reconfigurações geométricas significativas devido à fusão, compactação e remodelação do tecido (Mironov *et al.*, 2009-b). Ou seja, a estrutura bioimpressa é dinâmica e, portanto, seu projeto precisa prever este dinamismo.

A abordagem mais lógica para construir um órgão complexo como um rim, por exemplo, será primeiro especificá-lo virtualmente, utilizar modelos preditivos e realizar simulações *in silico*. Obviamente não será possível imprimir órgãos físicos sem um modelo digital e biológico e processamento de dados e informações (Rezende *et al.*, 2015).

A etapa crucial para a modelagem computacional é lançar o modelo em um formato padronizado e computável que possa ser analisado rigorosamente por meio de simulações e métodos matemáticos, enquanto as diferentes representações de modelos são úteis para diferentes fins (Dernowsek *et al.*, 2017-b).

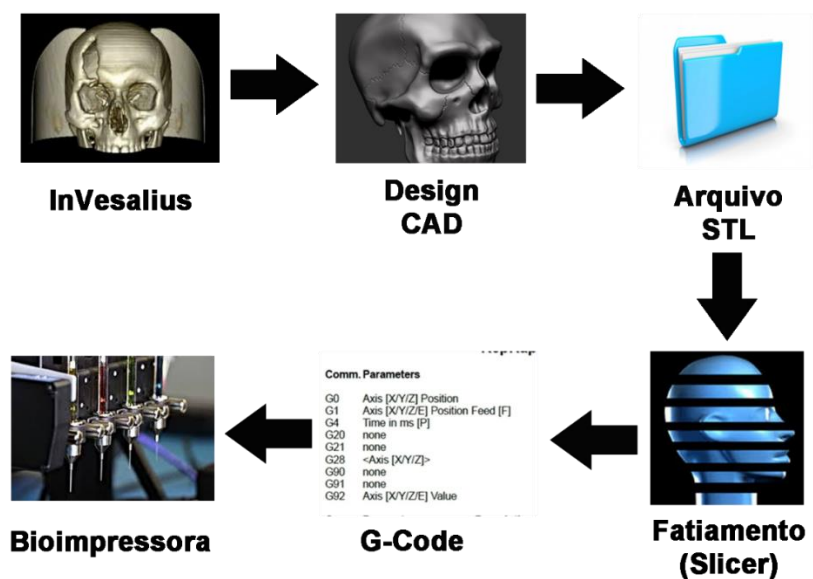
É essencial desenvolver um novo campo computacional que possa apoiar o projeto, modelagem, previsão, otimização, integração e tomada de decisão, e que possa auxiliar em todas as etapas do processo de biofabricação. Existem muitos grupos de pesquisa em todo o mundo tentando desenvolver ferramentas utilizando TI para compreender sistemas biológicos a fim de construir tecidos funcionais. Em particular, de uma perspectiva anatômica, as novas ferramentas devem ser capazes de integrar detalhes em níveis micro e macro do tecido ou órgão desejado a ser bioimpresso (Dernowsek *et al.*, 2017-a).

Iniciativa como o trabalho de Dávila *et al.* (2016) que desenvolveram o BioScaffolds PG, *software* que calcula e gera o caminho de impressão 3D para *scaffolds* de engenharia de tecidos (TE) que pode ser exportado para o *software* da impressora 3D, é essencial para o desenvolvimento da área da biofabricação.

Um sistema é definido como um conjunto de partes coordenadas que trabalham concorrente e sincronamente para realizar uma determinada tarefa, sendo este um todo organizado e complexo. Seguindo esta definição, o sistema computacional (SC) é um conjunto de dispositivos eletrônicos (*hardware*) capazes de processar dados de acordo com um programa (*software*). O principal *software*, e o mais importante, é o sistema operacional (SO), pois é ele quem “dá vida” ao *hardware* e fornece as bases para a execução de programas. Temos como exemplo de SO: *Windows, Linux, Unix, Snow Leopard (Apple)* (Fustinoni *et al.*, 2013).

No universo geral de TI, *hardware* é o aspecto físico de computadores, telecomunicações e outros dispositivos. Focando os holofotes na biofabricação, isso se refere aos equipamentos ou dispositivos que são empregados em uma linha de biofabricação para a produção de um tecido ou órgão. A linha de biofabricação inclui principalmente os seguintes equipamentos: classificadores de células, dispositivos microfluídicos, fabricantes e encapsuladores de esferoides de tecido, bioimpressoras, biorreatores, incubadoras entre outros (Rezende *et al.*, 2011). O *software*, de uma forma geral, tem a ver com as informações (instruções) que são processadas. A Figura 6 ilustra a sequência de etapas desde a concepção do projeto à biofabricação 3D.

Figura 6: Sequenciamento dos tipos de software usados no processo de bioimpressão.



Fonte: Produzida pelo autor.

1.5.1 DICOM - *Digital Imaging and Communications in Medicine*

Em 1983, a Associação Nacional dos Fabricantes de Equipamentos Elétricos (NEMA) e o Colégio Americano de Radiologia (ACR) formaram um comitê, denominado ARC-NEMA *Digital Imaging and Communications Standards Committee*, para desenvolver um padrão que permitisse que os equipamentos que utilizam imagens médicas digitais tais como: Tomógrafo Computadorizado, Raio-X, Ecógrafo, etc. pudessem comunicar-se entre si. Além de uma interface física, o padrão contemplava um dicionário dos elementos necessários na codificação e interpretação de imagens (Sampaio, 1999).

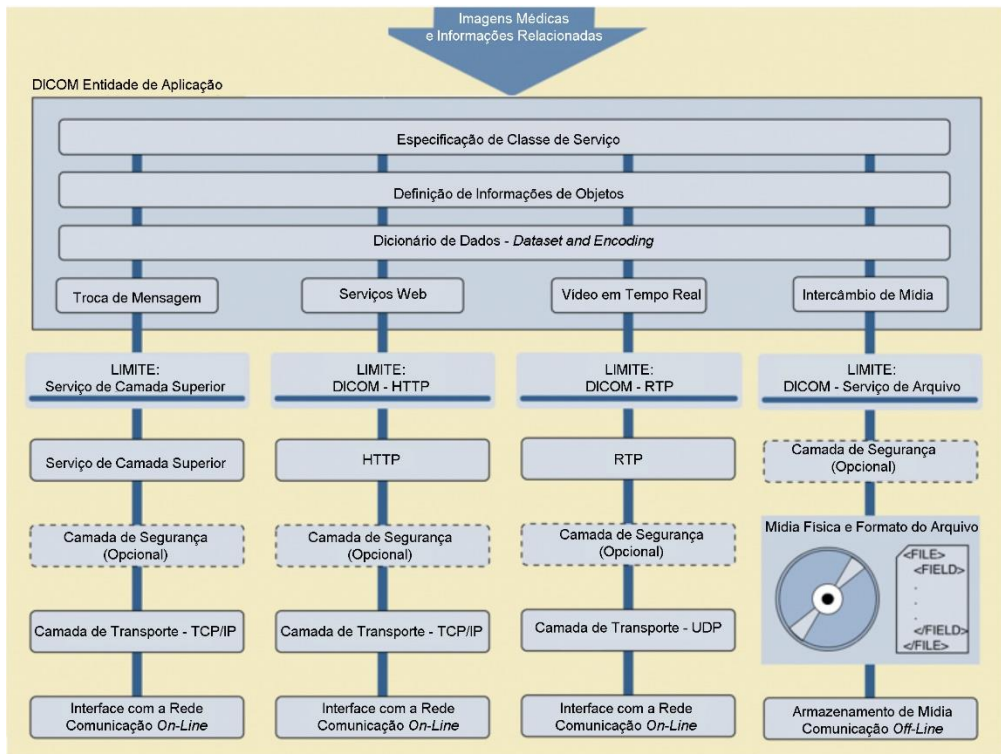
Em 1985, foi publicada a primeira versão do padrão *Digital Imaging and Communications in Medicine* – DICOM. Desde então, o padrão DICOM vem sendo aprimorado. O DICOM 3.0 pode ser aplicado a um ambiente em rede e à troca de mídia *off-line*, definindo o formato do arquivo e dando suporte a vários tipos de mídias físicas (Dicom Part 1, 2020). O padrão DICOM é administrado pela *Medical Imaging & Technology Alliance* - uma divisão da *National Electrical Manufacturers Association* (Dicomstandard, 2020). As principais definições do padrão DICOM são descritas no Apêndice A.

O padrão DICOM prevê um modelo de comunicação do padrão que contempla a troca de informações através da rede e armazenamento de arquivos, deste modo as entidades de aplicativo DICOM podem utilizar um dos três mecanismos de transporte:

- **Camada Superior de Serviço ou Serviço de Mensagens DICOM:** fornece independência de suporte específico de comunicação em rede física.
- **API DICOM de Web Service e HTTP Service:** permite o uso de hipertexto comum e protocolos associados para o transporte de serviços DICOM.
- **Serviço de Arquivos DICOM:** fornece acesso a mídia de armazenamento independentemente de formato específico de armazenamento da mídia física e de estrutura de arquivos.

A Figura 7 ilustra a arquitetura do modelo de comunicação previsto no padrão DICOM 3.0.

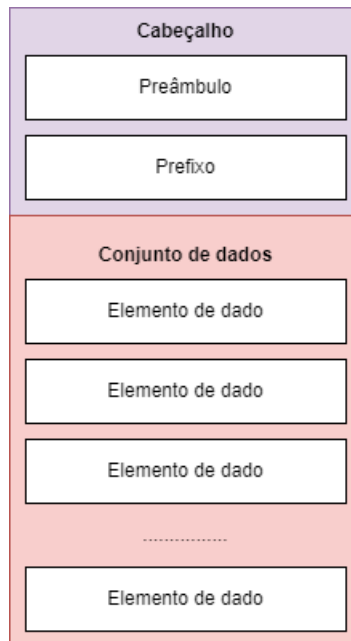
Figura 7: Modelo geral de comunicação DICOM



Fonte: Adaptado de DICOM Part 1, 2020.

A estrutura básica de arquivo DICOM descrita para o Armazenamento em mídia e formato de arquivo é apresentante na Figura 8 abaixo:

Figura 8: Estrutura Básica de um arquivo DICOM.



Fonte: Adaptado de Sampaio (1999).

O arquivo DICOM possui um cabeçalho composto por um preâmbulo de 128 bytes e um prefixo de 4 bytes, que pode ser suprimido do arquivo caso não seja necessário, seguido pelo conjunto de dados (DICOM Part 10, 2020).

Como visto o DICOM é um padrão dinâmico e muito bem estruturado, por isso se tornou unanimidade entre os fabricantes de equipamentos na área médica sendo usado na comunicação e geração de imagens médicas compatíveis com a maioria dos equipamentos.

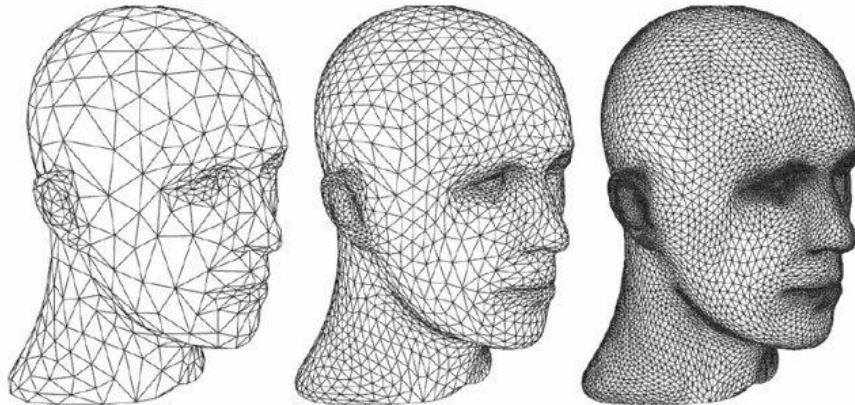
1.5.2 Arquivo STL

Um arquivo STL é resultado da conversão do modelo 3D em uma infinidade de triângulos, que representam cada pedaço do modelo, para possibilitar que as impressoras 3D materializem o projeto na sequência. A representação por triangularidade justifica-se pelo fato de ser a figura geométrica mais próxima de um vetor. Outro fator é que o triângulo possui propriedades como intensidade, direção e sentido, fundamentais para a impressão 3D (Grimm, 2004). *“Esse formato aproxima as superfícies de um modelo sólido com triângulos”* (3dsystems, 2020).

A sigla STL deriva de algumas letras da palavra *STereoLithography* (estereolitografia), com o passar do tempo esse termo passou a ter diferentes significados, tais como *“Standard Triangle Language”* (Linguagem Padrão do Triângulo) ou *“Standard Tessellation Language”* (Linguagem de Tesselação Padrão) (Grimm, 2004). *“A grande vantagem do STL para a tecnologia é que ele é um formato de arquivo universal e todas as impressoras 3D podem lê-lo”* (Grimm, 2004).

Conforme ilustrado na Figura 9, a criação do arquivo STL consiste na transformação da superfície do objeto em formas geométricas, normalmente triângulos.

Figura 9: Codificação da superfície do objeto em triângulos.



Fonte: 3DLAB, 2020.

O arquivo STL pode armazenar as informações das formas geométricas triangulares na codificação binária ou em ASCII, as quais gravam as informações do vetor bem como as coordenadas dos vértices (Grimm, 2004; Volpato, 2007).

- **Codificação ASCII:** a primeira linha é uma linha de descrição que deve começar com a palavra "sólido" em minúsculas, então normalmente contém o nome do arquivo, autor, data etc. A última linha deve ser a palavra-chave "fim sólido". As linhas entre as anteriores contêm descrições de 3 facetas de vértices incluindo seus normais, a ordem dos vértices deve obedecer à regra da mão direita. (Stl *Format*, 1999).
- **Codificação Binária:** Os arquivos STL binários consistem em uma linha de cabeçalho de 80 *bytes* que pode ser interpretada como uma *string* de comentário. Os seguintes 4 *bytes* interpretados como um inteiro longo fornecem o número total de facetas. O que se segue é um normal e 3 vértices para cada faceta, cada coordenada representada como um número de ponto flutuante de 4 *bytes* (12 *bytes* no total). Há um espaçador de 2 *bytes* entre cada faceta. O resultado é que cada faceta é representada por 50 *bytes*, 12 para o normal, 36 para os 3 vértices e 2 para o espaçador (Stl *Format*, 1999).

1.5.3 Arquivo AMF

O padrão mais recente para representação de produtos MA é o *Additive Manufacturing File Format* (AMF), uma iniciativa da *Sociedade Americana de Testes e Materiais* (ASTM). Desde 2013, o AMF foi adotado pela Organização Internacional de Padronização (ISO) como o padrão internacional ISO / ASTM 52915 (Iso/Astm, 2013) para substituir o padrão *STereoLitography* criado há 30 anos (Dernowsek *et al.*, 2017-a; Volpato, 2017).

Devido à limitação do arquivo STL, que contempla apenas as informações sobre a malha da superfície, não abrangendo informações como, texturas, materiais, subestruturas e cores, e ao avanço das tecnologias relacionadas à MA. Surgiu a necessidade de um arquivo que fosse capaz de conter informações além da malha de superfície (Volpato, 2007).

Nesse contexto foi criado o padrão AMF - *Additive Manufacturing File Format* da ASTM. Esse padrão de arquivo de fabricação aditiva suporta nativamente especificações de cores, materiais, treliças e constelações. Em seu formato eletrônico utiliza um esquema de XML (*eXtensible Markup Language*) (ASTM, 2012).

A primeira linha do arquivo AMF contém a declaração do XML bem como sua versão XML e codificação. O elemento root do arquivo é a TAG `<AMF></AMF>` (Volpato, 2007; Volpato,2017).

Segundo Lipson (2014), um documento AMF pode utilizar cinco elementos XML após o elemento raiz, mas somente o elemento `<object>` é obrigatório. A seguir serão descritos esses elementos:

- **<object>** - esse elemento define um ou mais volumes de materiais. Um identificador (ID) deve ser associado a cada volume para sua impressão;
- **<material>** - esse elemento é opcional, sua função é associar um material a um ID. Caso não exista nenhum elemento material, um único material padrão será utilizado;
- **<texture>** - esse elemento é opcional, sua função é mapear as imagens ou texturas usadas na impressão;
- **<constellation>** - esse elemento é opcional, sua função é combinar hierarquicamente os objetos e outras constelações em um padrão relativo utilizado para a impressão;

- **<metadata>** - esse elemento é opcional, sua função é especificar as informações adicionais referente aos objetos e os elementos contidos no arquivo “.AMF”.

1.5.4 *InVesalius*: SW livre de imagens médicas

O *InVesalius* é um SW livre desenvolvido pelo Centro de Tecnologia da Informação Renato Archer (CTI). O *software* utiliza imagens médicas provenientes de equipamentos de tomografia computadorizada ou ressonância magnética para reconstrução em 3D, permitindo a navegação nas fatias das imagens do exame através das orientações axial, coronal e sagital (CTI, 2020).

De acordo com o website do CTI, o *InVesalius* é usado, não somente na área médica, mas também nas áreas de ensino, análises forenses, veterinária, paleontologia e na indústria para inspeção de peças. É possível utilizá-lo nas plataformas *Microsoft Windows*, *GNU/Linux* e *Apple Mac OS X* (CTI, 2020).

O *InVesalius* permite a importação de arquivos no formato DICOM ou *Analyze*. Após o carregamento do arquivo, o *software* converte as imagens em uma matriz tridimensional armazenando essas informações em um arquivo *memmap* em disco e mapeadas em memória RAM (*Random Access Memory*) por meio da chamada de sistema *mmap*, comum a vários sistemas operacionais (Francisco, 2016).

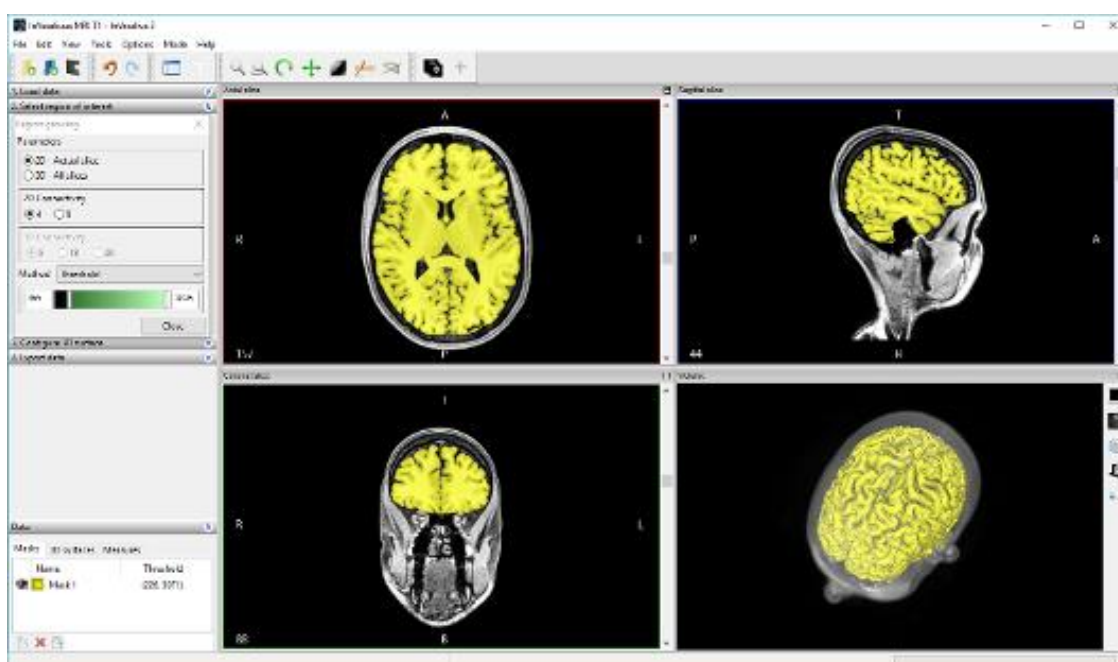
Segundo Amorim *et al.* (2011), o *InVesalius* utiliza o paradigma de orientação a objetos e foi desenvolvido utilizando a linguagem de programação *Python*. Outra característica do *software* é que ele permite o desenvolvimento de *plugins* pela comunidade. Segundo o website do CTI, além do Brasil, o *InVesalius* possui um histórico de utilização de 144 países ao redor do mundo, incluindo universidades hospitais de renome e grande indústrias (CTI, 2020).

O *InVesalius* permite a segmentação de imagens através de diferentes técnicas, como o *threshold*, *watershed* e de forma manual. Na primeira técnica apenas os *voxels* com valores entre um mínimo e um máximo são selecionados. Na segmentação por *watershed*, o usuário demarca objetos e detalhes do fundo. A técnica manual possibilita selecionar ou eliminar, interativamente, *voxels* de interesse do usuário não foram selecionados pela técnica de *threshold* (Amorim, 2011; Francisco, 2016).

Após a segmentação é possível reconstruir em 3D a área segmentada e utilizar as técnicas de MA para imprimir o modelo fisicamente. Também é possível fazer a visualização volumétrica e utilizar ferramentas de medidas linear e angular tanto em fatias 2D quanto na reconstrução 3D (Amorim *et al.*, 2011).

A Figura 10 demonstra uma Imagem de ressonância magnética com o cérebro selecionado com a ferramenta de *region growing*.

Figura 10: Imagem de ressonância magnética com o cérebro selecionado com a nova ferramenta de *region growing*.



Fonte: CTI, 2020.

1.5.5 Slicer (Fatiamento)

Segundo Volpato (2007), na MA o fatiamento é a etapa de maior importância no processo de planejamento, pois é nessa etapa que o objeto tridimensional é dividido em camadas de contornos, as quais servirão de guia para a impressão do objeto. O processo de fatiamento consiste em interseccionar a superfície do modelo 3D em camadas em diferentes alturas. Para aprimorar o processo de fatiamento podemos variar as seguintes características:

- Variação da altura entre as camadas;
- Fatiamento em diferentes representações;
- Ajuste de pontos de intersecção.

No processo de fatiamento, calcular o tamanho da camada pode reduzir o tempo de impressão, porém é preciso ter cuidado pois tamanhos inadequados podem comprometer a qualidade da superfície (Volpato, 2007; Volpato, 2017).

Diferentes abordagens são usadas no fatiamento (Figura 11). A abordagem Uniforme utiliza um tamanho uniforme para todas as camadas da peça. Já a abordagem Adaptativa possui variações de tamanhos ao longo do modelo tridimensional, conseguindo diminuir o tempo de impressão, mas com perda de fidelidade em alguns locais da peça. Por fim a abordagem Adaptativa Local, subdivide o modelo em partes verticais e aplica o fatiamento com camadas de tamanhos diferentes (Volpato, 2007; Volpato, 2017).

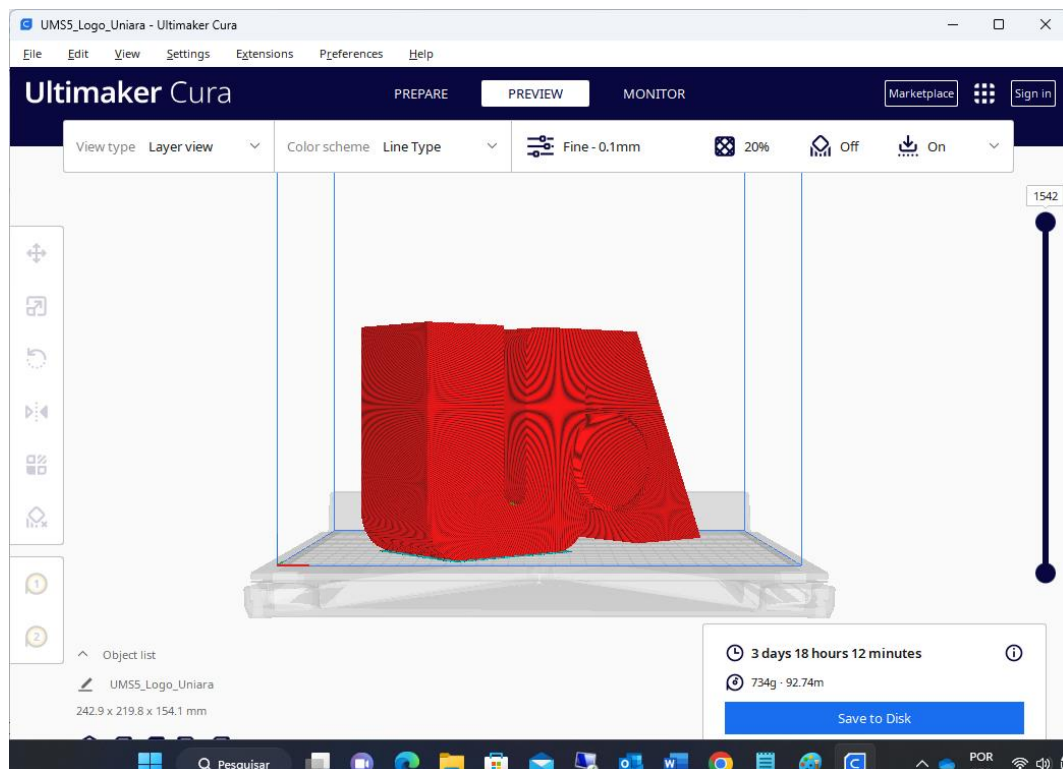
Figura 11: Abordagens no fatiamento.



Fonte: BUSTAMANTE, 2019.

A Figura 12 mostra a interface do software de fatiamento *Ultimaker Cura* com uma imagem fatiada.

Figura 12: Imagem fatiada no Ultimaker Cura.



Fonte: Produzida pelo autor.

1.5.6 gCode

O *gCode* é uma linguagem de programação controlada numericamente composta de uma série de comandos. Esses comandos informam à impressora 3D exatamente quais ações executar como, por exemplo: onde mover, que velocidade usar, quais temperaturas dos bicos e da bandeja (Simplify3d, 2020). *gCodes* são usados para uma variedade de máquinas, não apenas impressoras 3D. Ferramentas de corte, tornos e fresas também utilizam a linguagem *gCodes* (Bcn3d, 2020).

O *gCode* pode ser criado manualmente, por Programação Parametrizada, ou através da importação de um arquivo STL para um *software* de fatiamento. O *software* de fatiamento irá então configurar um código com base nas especificidades do *design* de sua impressão 3D. O código é então enviado para a impressora 3D para a construção física do arquivo digital (Bcn3d, 2020).

Cada linha de código do arquivo *gCode* diz à impressora para executar uma ação específica. Cada linha do arquivo *gCode* é composta de várias letras

e números diferentes com significados ou instruções específicas. Os comandos referentes à geometria ou que se referem diretamente à impressão do objeto, iniciam com a letra G. Os comandos não geométricos começam com M. Um exemplo de comando não geométrico poderia ser definir a temperatura do leito ou da extrusora (Bcn3d, 2020). A seguir um exemplo de comando *gCode*:

```
56 G1 F2200 X22.3 Y56.4 E5.9
```

O primeiro número indica o número da linha no arquivo. Cada linha do arquivo contém um comando. Essa contagem começa sempre em 1. Em seguida podemos ter as letras G ou M seguido por um número que indica uma ação específica que a impressora deve realizar. Para a maioria das impressoras, essas combinações de letras e números são universais (Bcn3d, 2020).

Em seguida, pode-se ter uma combinação com as seguintes letras:

- X seguido por um número e Y seguido por um número: Essas combinações de letras e números indicam as coordenadas X e Y de onde a cabeça de impressão deve ir. Alguns *gCodes* também podem incluir uma coordenada Z.
- F seguido por um número indica a taxa de avanço. Basicamente, isso informa à impressora a velocidade de movimentação do cabeçote de impressão.
- E seguido por um número indica quanto filamento será descarregado da extrusora.

Existem algumas letras e números adicionais que são usados para códigos M. Por exemplo, T indica qual extrusora está sendo usada, começando com T0 e subindo com base no número de extrusoras. S indica o grau em Celsius para uma extrusora (Bcn3d, 2020).

O *gCode* é a última etapa antes de enviar os dados para a impressora 3D. Primeiramente, um projeto 3D é feito em um *software* CAD e exportado como um STL, o arquivo STL é então importado para um *software* de segmentação, como BCN3D Cura. No *software* de fatiamento são definidos os parâmetros para impressão e em seguida o *software* criará e exportará o arquivo com extensão *gCode*. Esse arquivo é então enviado para a impressora 3D por meio de um cartão SD, *wi-fi* ou interface conectada ao computador. A impressora então usa cada linha de código para cada ação que irá realizar (Bcn3d, 2020).

A Figura 13 mostra um exemplo de um código *gCode* exportado do *Ultimaker Cura* e aberto no *Visual Studio Code*.

Figura 13: Exemplo de código escrito em *gCode*.

```
1 |;Start GCode
2 |M109 S210.000000
3 |G28 X0 Y0 Z0
4 |G92 E0
5 |G29;Layer 1 of 20
6 |;perimeter
7 |G0 F2700 X74.72597295 Y73.8177037 Z-0.4304701
8 |G1 F900 X75.40090594 Y74.37373364 E0.043723673083546626
9 |G0 F2700 X74.73495632 Y73.8251045 Z-0.4304701
10 |G1 F900 X75.40090594 Y74.37373364 E0.08686538229966613
11 |G0 F2700 X74.72597295 Y73.8177037 Z-0.4304701
12 |G1 F900 X74.73495632 Y73.8251045 E0.08744734616709456
13 |G0 F2700 X74.76034443 Y73.84602 Z-0.4304701
14 |G1 F900 X75.40090594 Y74.37373364 E0.12894435611767804
15 |G0 F2700 X74.73495632 Y73.8251045 Z-0.4304701
16 |G1 F900 X74.76034443 Y73.84602 E0.13058905538321422
17 |G0 F2700 X74.79979365 Y73.8785195 Z-0.4304701
18 |G1 F900 X75.40090594 Y74.37373364 E0.16953045461478344
19 |G0 F2700 X74.76034443 Y73.84602 Z-0.4304701
20 |G1 F900 X74.79979365 Y73.8785195 E0.17208606533379875
21 |G0 F2700 X74.85096048 Y73.9206722 Z-0.4304701
22 |G1 F900 X75.40090594 Y74.37373364 E0.20771276449992573
23 |G0 F2700 X74.79979365 Y73.8785195 Z-0.4304701
24 |G1 F900 X74.85096048 Y73.9206722 E0.21102746456536864
25 |G0 F2700 X74.91150133 Y73.9705477 Z-0.4304701
26 |G1 F900 X75.40090594 Y74.37373364 E0.24273218718294312
27 |G0 F2700 X74.85096048 Y73.9206722 Z-0.4304701
28 |G1 F900 X74.91150133 Y73.9705477 E0.24665416373150015
29 |G0 F2700 X74.979072765 Y74.02621496 Z-0.4304701
30 |G1 F900 X75.40090594 Y74.37373364 E0.27398146054620187
31 |G0 F2700 X74.91150133 Y73.9705477 Z-0.4304701
32 |G1 F900 X74.979072765 Y74.02621496 E0.27835888634907585
```

Fonte: Produzida pelo autor.

2 Justificativa

A área de Biotecnologia entende que a associação de conhecimentos de outras áreas, como a informática, e a engenharia no desenvolvimento de pesquisas e produtos deverá ser percebido como ferramentas à geração de resultados inovadores, porém que estas não sejam um fim em si mesmo, descaracterizando o viés biotecnológico essencial do processo (Capes, 2019).

Neste projeto, a TI é utilizada como uma ferramenta (meio) para beneficiar a área de Bioimpressão 3D contribuindo para a obtenção de estruturas vivas estruturadas e funcionais aplicáveis à MR.

A bioimpressão 3D, como uma área ainda muito recente da C&T, assim como qualquer outro campo, possui suas grandes barreiras a serem transpostas.

Os principais obstáculos para a potencial biofabricação de tecidos e órgãos humanos funcionais transferíveis ao paciente compreendem:

- *Blueprint (design)* complexo da anatomia a ser impressa considerando fatores como a evolução da estrutura em relação ao tempo.
- Garantia da vascularização dos tecidos neoformados;
- Eliminação ou redução de rejeição pelo paciente transplantado;
- Desenvolvimento de biomateriais (inteligentes);
- Suprimento de material biológico em larga escala;
- O controle do tempo (4D) na estrutura viva em formação;
- Dispositivos, como os biorreatores, apropriados para a maturação do órgão;
- Criação de modelos matemáticos e computacionais preditivos de processos e fenômenos;
- Aspectos burocráticos de regulamentação;
- Aspectos éticos.

Todos esses itens representam enormes desafios e que já vêm sendo desenvolvidos pela comunidade científica internacional. O item relacionado ao *design* do órgão, é o desafio mais diretamente relacionado a este projeto. As TIs, que envolvem uma séria de ferramentas e métodos correlacionados a *software*

(SW) e *hardware* (HW) para a bioimpressão, são parte do cotidiano da bioimpressão como os próprios biomateriais e células.

Pesquisas precisam ser realizadas a fim de otimizar a tradução de um projeto digital da estrutura anatômica alvo em um arquivo CAD para impressão, incluindo componentes de sacrifício e de construção. Além disso, o SW e o HW devem ser integrados para a fabricação reproduzível com várias ferramentas (Visser *et al.*, 2013).

Existem diversos SW utilizados na bioimpressão desde a etapa inicial à etapa final. Existem SW embarcados nos equipamentos de aquisição de imagens médicas, por ressonância magnética ou tomografia computadorizada, SW desenvolvidos para a montagem da anatomia 3D digital bem como a segmentação daquelas imagens, SW para a preparação do arquivo (normalmente STL) que será formado e fatiado para ser levado à bioimpressora, além do próprio SW embarcado nas bioimpressoras. Sem falar, da etapa de maturação e outros SW auxiliares e complementares do processo, ou mesmo pacotes de SW utilizados em simulações computacionais.

Há uma escassez de ferramentas de SW especializadas adequadas. É necessário desenvolver uma nova área de aplicação, como a ciência da computação para bioimpressão, que pode contribuir significativamente em pesquisas futuras (Pakhomova *et al.*, 2020).

Ainda não existe uma representação ou método que crie um *design* complexo contendo, por exemplo, detalhes micrométricos da estrutura anatômica, como a vascularização, e considerando o processo dinâmico da sua formação.

Atualmente, a bioimpressão de qualquer estrutura anatômica, em nível de pesquisa, tem sido realizada pelos cientistas com base na literatura por meio de artigos publicados (Long Ng *et al.*, 2018) que tragam pistas mínimas em termos do planejamento adotado para a criação de determinada anatomia em laboratório. No entanto, as publicações nem sempre são completamente reproduzíveis, pela ocultação de determinadas informações sigilosas, obrigando os cientistas a procederem em um processo de tentativa-e-erro na adaptação dos parâmetros e características para seus próprios casos. Estes parâmetros e características envolvem, por exemplo, o(s) tipo(s) de células utilizados na biotinta, bem como, a estrutura anatômica almejada, além da forma como as

camadas com biotintas serão depositadas e, principalmente, a resposta celular com a diferenciação das camadas a serem verificadas nos dias subsequentes.

Estudar e desenvolver a bioimpressão 3D multimaterial e desenvolver novas abordagens neste campo de pesquisa representam um dos elementos críticos que farão progresso no campo da bioimpressão (Pakhomova et al., 2020).

Ao mesmo tempo que não há um *blueprint*, há também uma grande dificuldade por parte dos usuários em compreender as etapas da bioimpressão relacionadas à TI e seu uso. Muitas vezes isso é motivo de desmotivação e uma barreira permanente ao aprofundamento de estudos e análises e, naturalmente, à obtenção de melhores resultados.

Dada a interdisciplinaridade da área, por exemplo, biólogos podem ter menos afinidade com o uso de SW e HW e pouco tempo e interesse em se dedicar a aprender ferramentas e metodologias de SW aplicadas ao processo de bioimpressão. Paralelamente, o engenheiro tem dificuldades e limitações em sua formação para interpretar dados biológicos como testes de viabilidade celular, por exemplo. O estreitamento no canal entre os diferentes perfis participantes do processo é fundamental para acentuar a comunicação e interatividade entre eles possibilitando melhores resultados. Assim, um SW de integração com interface facilitadora estimulará e acelerará o trabalho.

A proposta aqui apresentada oferece benefícios a qualquer usuário de qualquer formação que inevitavelmente fará uso de SW no processo de bioimpressão, pois estende as funções de SW utilizado para bioimpressão.

No caso da TI, esse projeto será focado em auxiliar o usuário no processo atual de bioimpressão com a facilitação de execução das etapas, integrando imagens, SW de *design* e de fatiamento, além da geração do *gCode* com parâmetros adicionais que reforçarão as informações passadas à bioimpressora e registradas do processo.

De forma resumida, será desenvolvida uma extensão acoplável ao *InVesalius* para que as etapas de fatiamento, geração do *gCode* e ajustes dos parâmetros do *gCode* possam ser feitos diretamente no *InVesalius* e, deste modo, o modelo gerado em 3D possa ser enviado diretamente para a impressora sem a necessidade de utilização de outros *softwares* intermediários. O foco do trabalho está no processo de impressão por extrusão de material.

Objetiva-se também criar uma forma para que os pesquisadores possam armazenar os parâmetros usados nos processos de biopressão. Após a validação dos parâmetros, pode-se armazenar, em arquivo, quais parâmetros foram mais eficientes para a biofabricação de um determinado tipo de tecido ou órgão.

Este projeto contempla uma contribuição na direção da integração dos SW e a criação de um arquivo final que incorpore parâmetros operacionais de processamento. Destina-se, portanto, a desenvolver uma solução computacional para a integração e interoperabilidade de SW utilizados no processo da bioimpressão, trazendo benefícios aos usuários como a abreviação de determinadas sub-etapas, economia no tempo do processo, facilitação de uso aos diferentes perfis de usuários que utilizarão a tecnologia e que, muitas vezes, não possuem conhecimento e intimidade com uso de SW específicos e facilitação na gestão de dados experimentais.

3 Objetivos

Desenvolver um *plug-in* para a integração e interoperabilidade dos *softwares* utilizados no processo de bioimpressão 3D, desde o uso de imagens médicas à materialização 3D, para a aplicação na área da Biotecnologia como uma ferramenta no processo de bioimpressão de tecidos e órgãos.

3.1 Objetivos Específicos

- Estender as funcionalidades do *InVesalius*, adicionando funcionalidades para o fatiamento e impressão de modelos para o processo de Bioimpressão;
- Permitir a exportação dos parâmetros e configurações do sistema para gestão de dados pesquisa/experimentos;
- Parâmetros *gCode*: incorporação de parâmetros;
- Validação computacional: realização de testes e apuração de *bugs* e correção.
- Registro de *software*.

4 Materiais e Métodos

Os materiais e métodos compreendidos no projeto são apresentados a seguir.

4.1 Impressora e Bioimpressora 3D

Foram utilizadas impressoras 3D e bioimpressora 3D para o desenvolvimento do projeto. As impressoras *Creality Ender 3 Pro*, *3D UP Plus 2 3D printer* e a *3DCloner ST DH G3* foram utilizadas para a impressão de estruturas não-vivas de arquivos *gCode* gerados no protótipo bem como para testar a comunicação com a impressora através do envio dos códigos G pela interface USB (*Universal Serial Bus* - Porta Serial Universal).

A bioimpressora 3D Modelo Genesis Vessels, série 2018/1, da fabricante 3DBS foi utilizada para a impressão – via porta USB - dos testes do *software* proposto. Foi realizada uma visita à startup 3DBS para um treinamento, em Campinas-SP.

4.2 Computador e sistema operacional de desenvolvimento

O trabalho de pesquisa e desenvolvimento na programação do SW integrador foi realizado com um *notebook Dell Inspiron 3501* com processador Intel I7 8 Gigabytes de memória RAM e um *notebook Acer modelo E5-571/E5-531* equipado com processador Intel I5 e 8 Gigabytes de memória RAM.

Devido à natureza do *InVesalius* ser multiplataforma e a linguagem de programação a qual o *InVesalius* foi escrito também ser multiplataforma, optou-se em utilizar o sistema Windows nas versões 10 e 11 como sistema operacional para o desenvolvimento do projeto.

4.3 *InVesalius*

O *InVesalius* é um *software* livre desenvolvido pelo Centro de Tecnologia da Informação Renato Archer (CTI), disponibilizado gratuitamente para

download através do website <https://www.cti.gov.br/invesalius>, baseado na linguagem de programação *Python*. O *software* permite a importação de arquivos no formato DICOM ou *Analyze*. Após o carregamento do arquivo, o *software* converte as imagens em uma matriz tridimensional permitindo a visualização das imagens em 3D e a segmentação, sendo possível reconstruir em 3D a área segmentada e utilizar as técnicas de MA para imprimir o modelo fisicamente. O modelo 3D gerado deve ser exportado em forma de um arquivo STL para que seja feito, na sequência, o processo de fatiamento e geração do *gCode* em outros *softwares*. Neste projeto foi utilizada a versão 3.1 do *InVesalius*.

4.4 *Visual Studio Code*

O *Visual Studio Code* é um editor de código-fonte que é executado na área de trabalho e está disponível para *Windows*, *MacOS* e *Linux*. Este editor vem com suporte integrado para *JavaScript*, *TypeScript* e *Node.js* e permite a adição de novos recursos através de extensões como, por exemplo, a inclusão de outras linguagens (como C ++, C #, *Java*, *Python*, *PHP*, *Go*) (Microsoft, 2020). Além de permitir a inclusão de extensões de outras linguagens, o *Visual Studio Code* possui licença MIT (*Massachusetts Institute of Technology* - Instituto de Tecnologia de *Massachusetts*), que permite o uso do *software* gratuitamente. Devido ao suporte à linguagem *Python* que foi usada no desenvolvimento do projeto e ao tipo de licenciamento do *Visual Studio Code* optou-se por usar esse *framework* para o desenvolvimento do projeto.

4.5 *Python*

No projeto foi usada a linguagem de programação *Python*. O principal motivo do uso dessa linguagem é que o *InVesalius* faz uso desta linguagem em sua programação. O uso da mesma linguagem facilita a integração da extensão que se pretende criar para o *InVesalius*. Outro motivo é que a linguagem *Python* possui diversas bibliotecas para tratamento de imagem que ajudam na elaboração do projeto. Neste trabalho, foram utilizados os seguintes pacotes/módulos do *Python* descritos na lista abaixo separados por arquivo *.py*.

- FrPrincipal.py
 - import wx
 - import wx.xrc
- app.py
 - import FrPrincipal
 - import tempFile
 - from assets.vtkViewer.vtkViewer import *
 - from
 - assets.vtkViewer.wxVTKRenderWindowInteractor.wxVTKRender
 - WindowInteractor import wxVTKRenderWindowInteractor
 - from assets.slicer.slicing import *
 - import serial.tools.list_ports as p
- tempFile.py
 - import numpy
 - from stl import mesh
 - import stl
 - import os
 - from datetime import datetime
 - import shutil
- slicing.py
 - import math
 - import sys
 - import string
 - import copy
- vtkViewer
 - import math
 - import wx
 - import vtk
 - import sys
 - import os
 - from
 - assets.vtkViewer.wxVTKRenderWindowInteractor.wxVTKRender
 - WindowInteractor import wxVTKRenderWindowInteractor

- import tempFile

5 Desenvolvimento do Trabalho

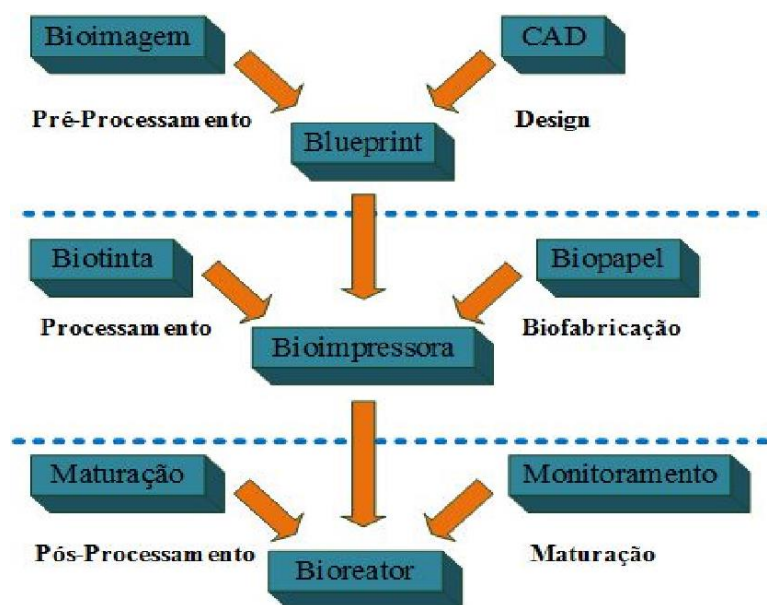
Conhecer as fases do processo de biofabricação ajuda a compreender como a integração dos *softwares* melhora o processo de biofabricação. No próximo parágrafo são apresentadas brevemente as fases do processo de biofabricação.

Segundo Mironov, Kasyanov e Markwald (2011), a complexidade da bioimpressão de tecidos vivos e de órgãos exige a utilização de uma série de dispositivos robóticos integrados automatizados ou uma linha de biofabricação. Mironov, Kasyanov e Markwald (2011) dividiram a biofabricação nas seguintes fases:

1. **Pré-processamento e *design*** - nesta fase é desenvolvido o projeto em 3D do tecido ou órgão que se deseja imprimir. O pré-processamento consiste na construção do modelo 3D em CAD (*Computer Aided Design*) a ser impresso. De acordo com Mironov *et al.* (2003), o modelo 3D pode ser obtido a partir de imagens de tecidos ou órgãos digitalizadas. A obtenção das imagens digitalizadas pode ser feita, por exemplo, por meio de exames de ressonância magnética ou tomografia computadorizada.
2. **Processamento e biofabricação** - nesta fase é feita a impressão do projeto elaborado na fase anterior. De acordo com Ozbolat e Yin Yu (2013), na fase de processamento as células e/ou biomateriais carregados de células são posicionados com precisão, e, então, servem como blocos de construção para a montagem de uma estrutura 3D organizada que, no futuro, poderá vir a ser um órgão funcional.
3. **Pós-processamento e maturação** - nesta fase o tecido impresso é maturado e monitorado. O monitoramento e a maturação do tecido são feitos em biorreatores. Segundo Martin, Wendt e Heberer (2004), os biorreatores permitem o controle rigoroso das condições ambientais e operacionais tais como pH, pressão, fornecimento de nutrientes para o material biológico e remoção de resíduos, viabilizando o monitoramento do material biológico bem como seu desenvolvimento.

A Figura 14 ilustra as fases existentes em um processo de biofabricação.

Figura 14: Fases de um processo de biofabricação.

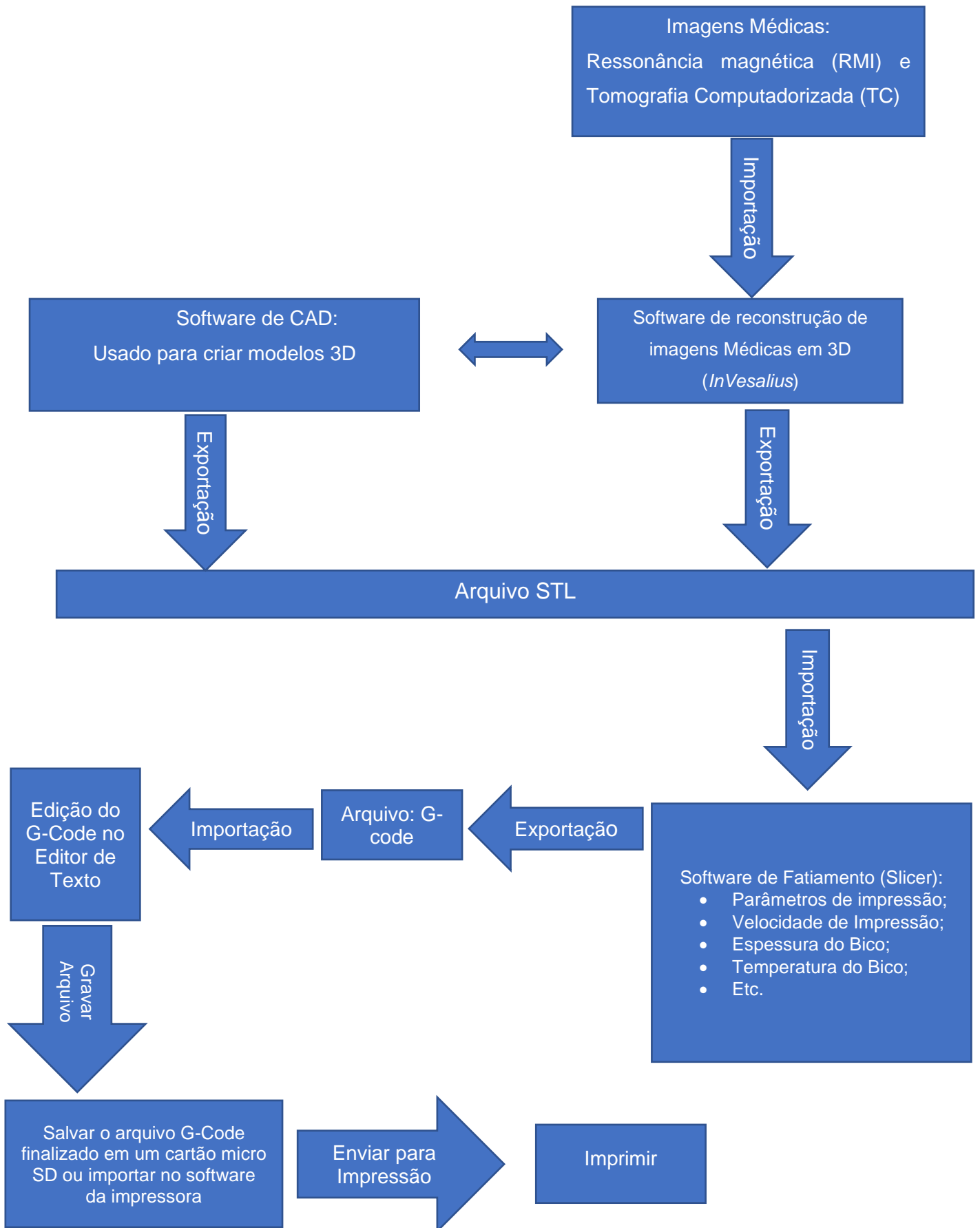


Fonte: Mironov *et al.*, 2011

O *plugin* desenvolvido nesse trabalho atua nas fases de pré-processamento e processamento, otimizando os processos da bioimpressão em algumas etapas das fases de pré-processamento e processamento propostas por Mironov, Kasyanov e Markwald (2011).

As etapas para a Bioimpressão 3D, em impressoras por extrusão, iniciam na obtenção do modelo a ser impresso, o qual pode ser obtido por meio de imagens médicas, por projeto feito em algum *software* CAD ou por meio de parametrização de geometrias utilizando programação (neste caso não precisa de fatiamento). Em seguida, o modelo 3D é convertido para o padrão de triangularização (stl) e exportado para um arquivo com a extensão stl. O arquivo stl é importado pelo *software* de fatiamento (*slicing*) onde os parâmetros de fatiamento devem ser definidos para, em seguida, o modelo ser fatiado camada a camada e o *gCode* ser gerado. Caso o *software* de fatiamento tenha recursos de edição do *gCode* e recurso de envio do *gCode* direto para a impressora o arquivo pode ser impresso, caso contrário, é necessário salvar o *gCode* em um arquivo, editá-lo se necessário, abrir o arquivo *gCode* no *software* da bioimpressora ou gravá-lo no cartão micro SD e enfim iniciar a impressão do modelo. A Figura 15 resume esse processo.

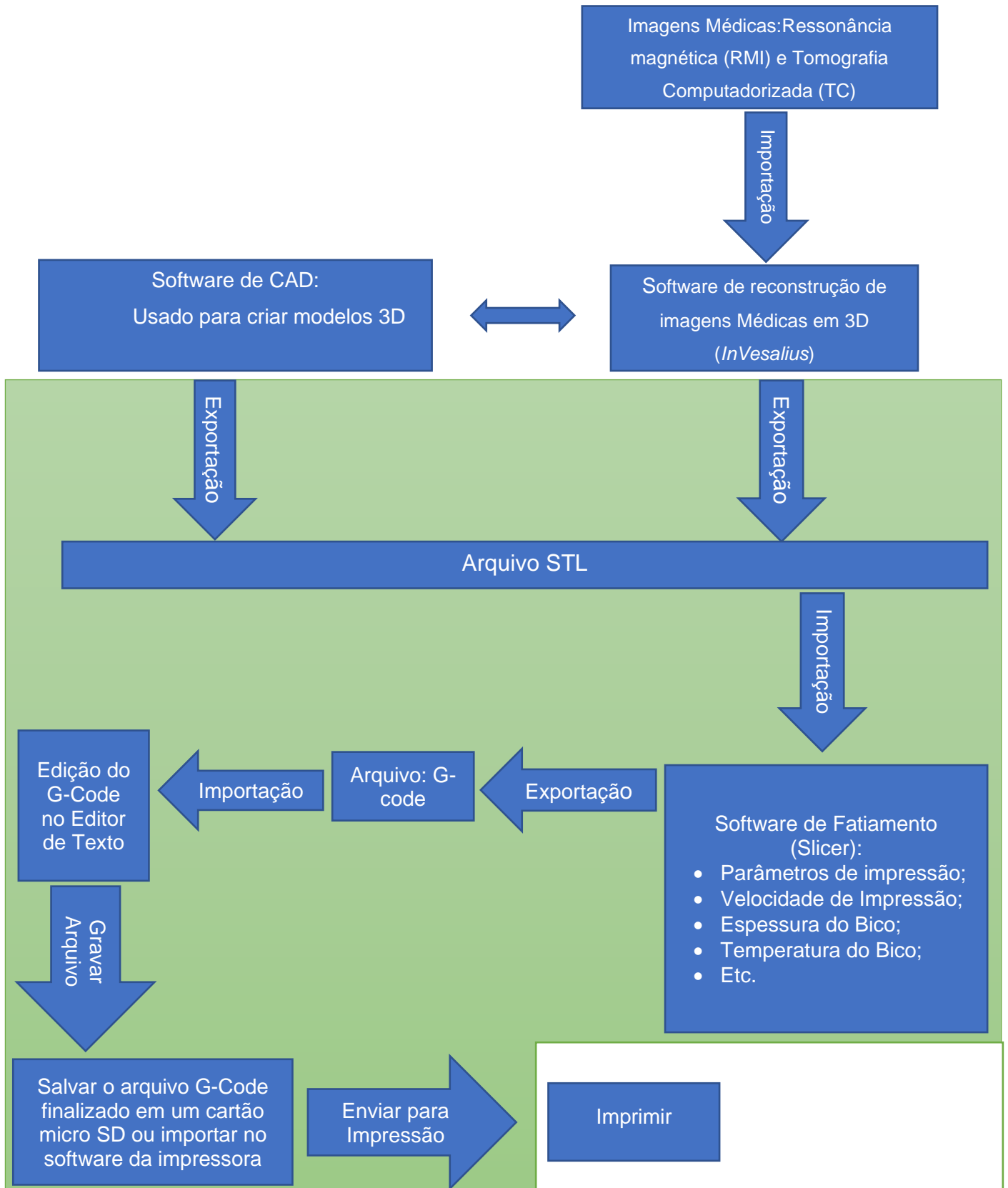
Figura 15: Fluxograma das etapas da Bioimpressão por extrusão.



Fonte: Produzida pelo autor.

Com o *plugin* instalado no *InVesalius*, o usuário não precisará instalar outros *softwares* para imprimir o modelo selecionado. O *plugin* permite ao usuário, após a seleção do modelo a ser impresso e a conversão em STL, clicar em um botão e os recursos necessários para realizar a visualização do modelo, parametrização, fatiamento, ajustes no *gCode* e envio do modelo para a bioimpressora estão disponíveis. Na Figura 16, as etapas da bioimpressão que são encapsuladas pelo *plugin* proposto neste trabalho são destacadas pela cor verde.

Figura 16: Etapas da bioimpressão encapsuladas pelo plugin.

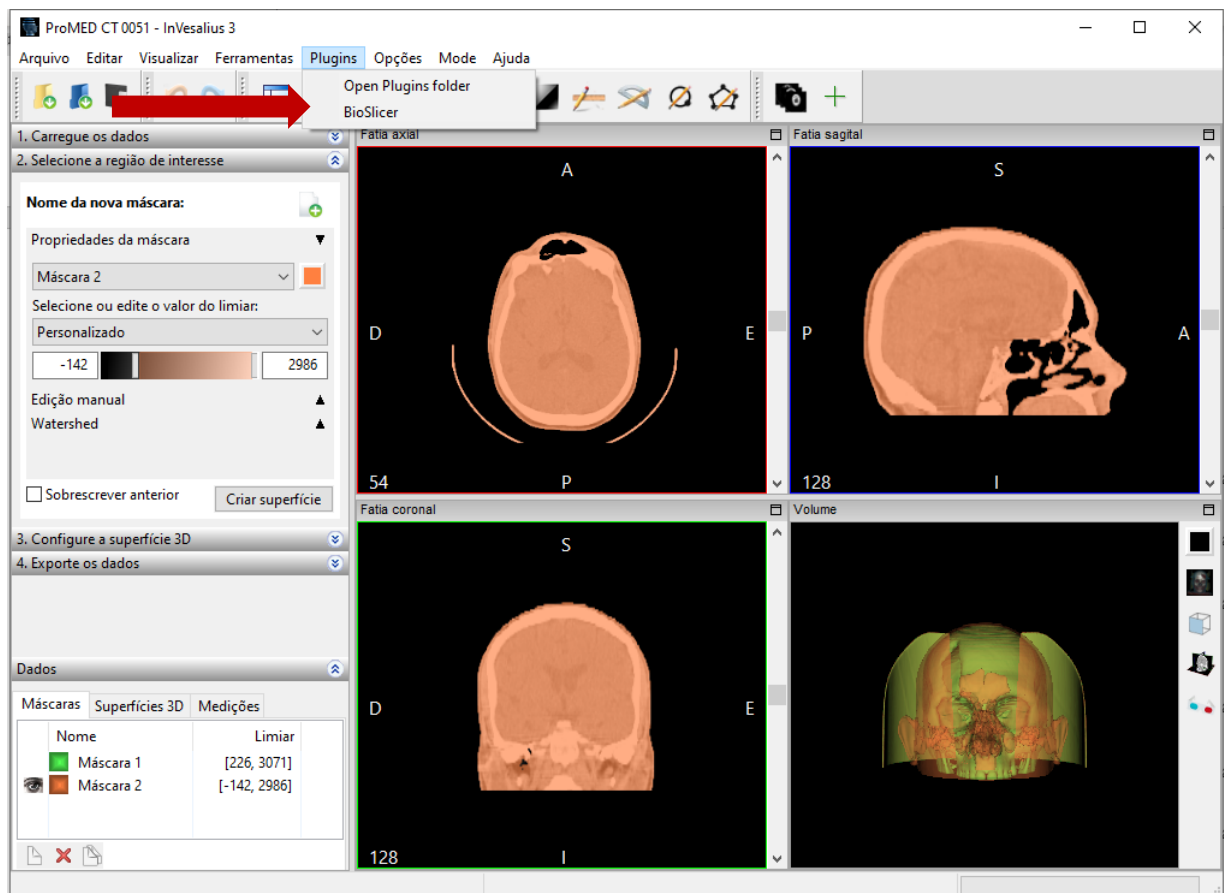


Fonte: Produzida pelo autor.

Conforme mencionado anteriormente o *plugin* permite ao usuário, após a seleção do modelo a ser impresso no *InVesalius* e a exportação do arquivo STL, clicar na opção *Plugins* no menu e escolher a opção *BioSlicer* e os recursos necessários para realizar a visualização do modelo, parametrização, fatiamento, ajustes no *gCode* e envio do modelo para a bioimpressora estarão disponíveis para o usuário. A Figura 17 mostra como acessar *plugin BioSlicer* através do menu do *InVesalius*.

O *plugin* foi registrado no INPI com o nome de *BioSlicer* (vide documento de registro nos anexos). O *BioSlicer* está disponível em duas versões: Uma para ser instalado como *plugin* no *InVesalius* e outra autônoma, que permite ao usuário usar o *software* independente do *InVesalius* estar instalado no computador do usuário.

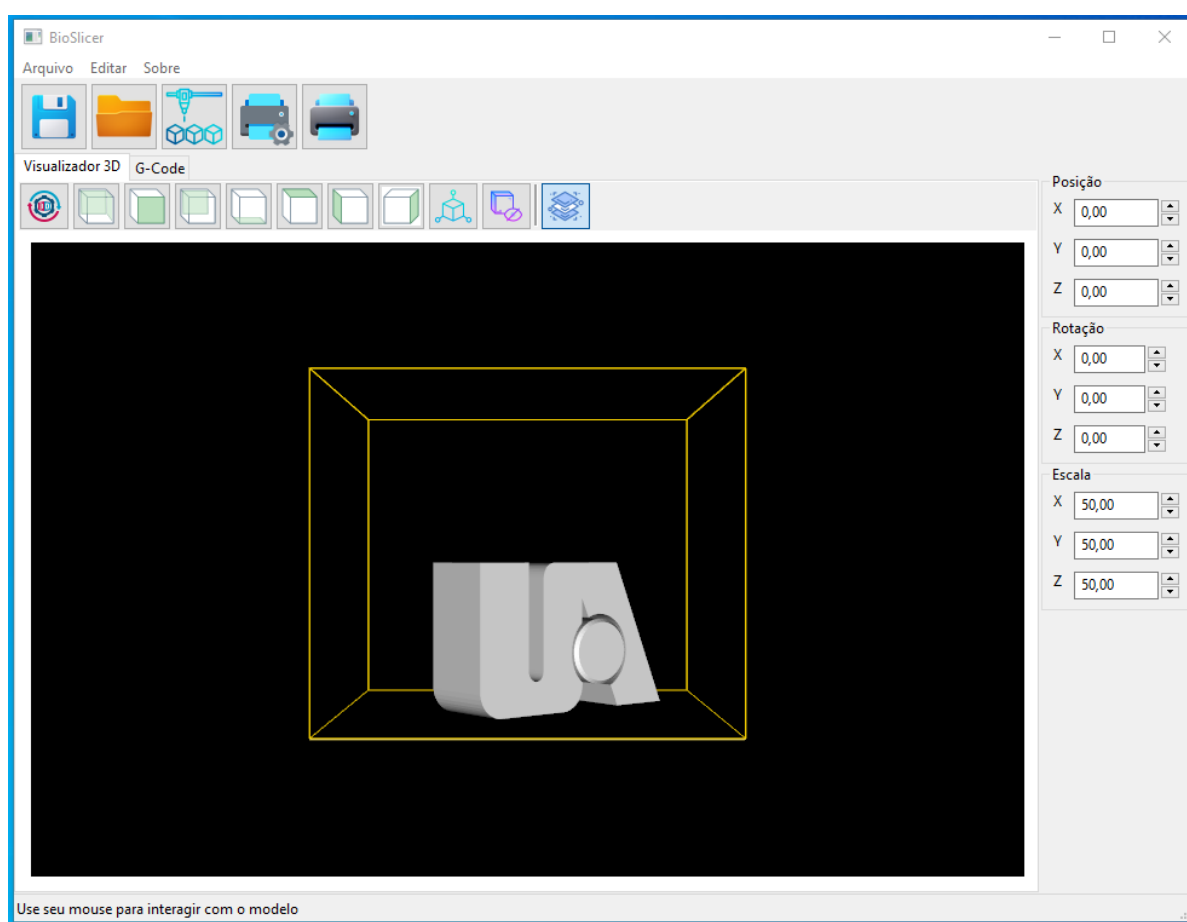
Figura 17: Passos para acessar o plugin BioSlicer no InVesalius.



Fonte: Produzida pelo autor.

Ao clicar na opção *BioSlicer* do menu *Plugins* do *InVesalius* o software *BioSlicer* é aberto. A Figura 18 ilustra a interface do *plugin BioSlicer*.

Figura 18: Interface do Plugin BioSlicer.



Fonte: Produzida pelo autor.

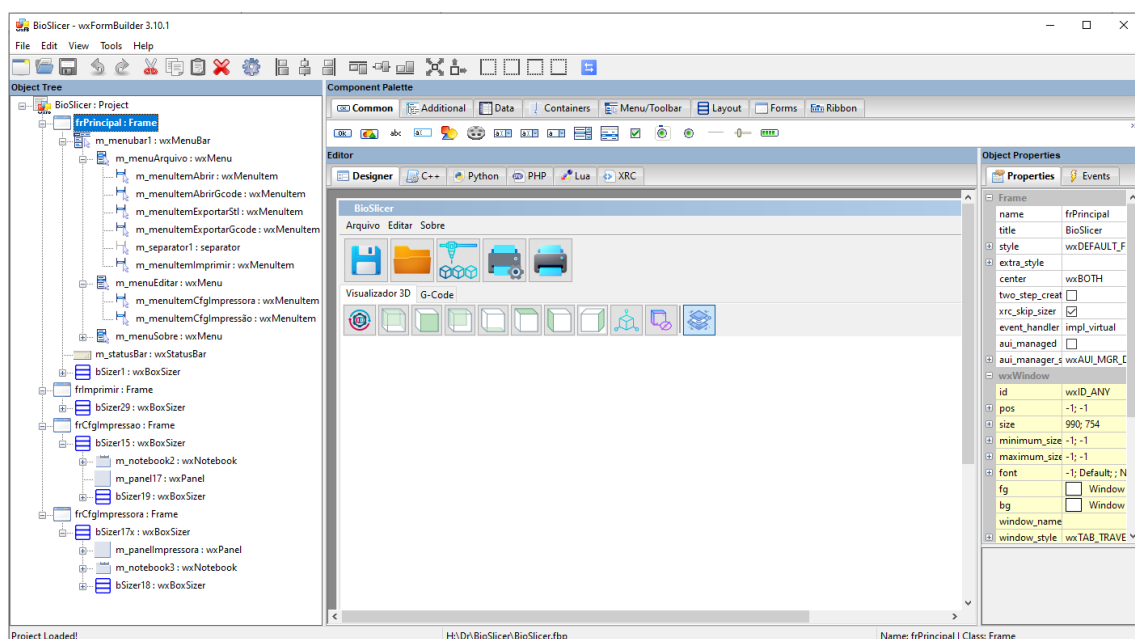
5.1 Desenvolvimento da Interface do *BioSlicer*

Para o desenvolvimento da interface do *BioSlicer* foi usada a biblioteca *wxPython*. O *wxPython* é um conjunto de ferramentas GUI para a linguagem de programação *Python*, ele permite que os programadores *Python* criem programas com uma interface gráfica de usuário robusta e altamente funcional.

O *wxPython* é implementado como um conjunto de módulos de extensão *Python* que envolvem os componentes da GUI da biblioteca *wxWidgets*, escrita em C++ (Wxpython, 2023).

Para agilizar o desenvolvimento da interface do *plugin* foi utilizado o *wxFormBuilder*. O *wxFormBuilder* permite desenvolver a interface graficamente, clicando e arrastando os componentes, e depois gerar o código da interface em *Python*.

Figura 19: Projeto da interface visualizado no wxFormBuilder.



Fonte: Produzida pelo autor.

A Figura 19 mostra o projeto da interface do *plugin*. No lado esquerdo da figura estão os componentes usados. O *FrPrincipal* é um componente do tipo *Frame*. Um *Frame* é uma janela cujo tamanho e posição podem (geralmente) ser alterados pelo usuário. Eles são posicionados hierarquicamente para que seja mantido o alinhamento dos componentes. Dentro do *Frame* principal temos os seguintes componentes *wxBoxSizer*, *wxPainel*, *wxStaticBoxSizer*, *wxTextCtrl*, *wxStaticText*, *wxButton*, *wxMenuBar*, *wxMenu*, *wxMenuItem* e *wxStatusBar*.

A função dos componentes usados na construção da interface é descrita abaixo:

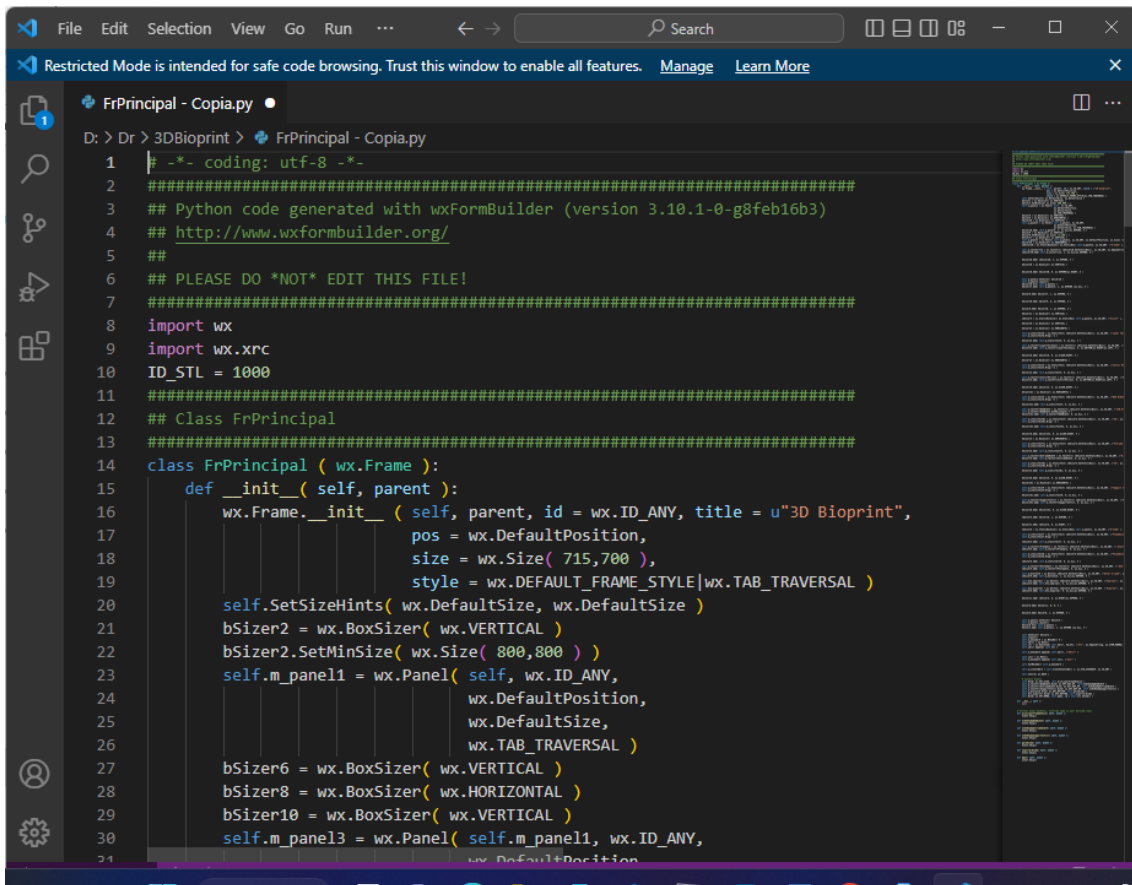
- **wxBoxSizer:** O *wxBoxSizer* é usado para posicionar outros componentes dentro dele, geralmente em uma linha ou coluna ou em várias hierarquias encadeadas;
- **wxPainel:** Um painel é uma janela na qual os controles são colocados;
- **wxStaticBoxSizer:** O *wxStaticBoxSizer* é um dimensionador derivado de *wxBoxSizer*, mas adiciona uma caixa estática ao redor do dimensionador;
- **wxTextCtrl:** permite que o texto seja exibido e editado;
- **wxStaticText:** exibe uma ou mais linhas de texto somente leitura;
- **wxButton:** exibe um botão na tela. Ao clicar no botão um evento é disparado;
- **wxMenuBar:** é uma série de menus acessíveis na parte superior de um quadro (*Frame*);
- **wxMenu:** é uma lista de itens suspensa ou pop-up, que pode ser selecionado antes que o menu desapareça;
- **wxMenuItem:** representa um item em um menu;
- **wxStatusBar:** é uma janela estreita que pode ser colocada na parte inferior de um quadro para fornecer pequenas quantidades de informações de status.

Ao centro da Figura 19 é exibida a interface que está sendo construída graficamente. Através dessa tela é possível verificar o posicionamento e alinhamentos dos componentes sem a necessidade de compilar o código-fonte.

Na direita da Figura 19 estão as propriedades e eventos de cada componente. Através dessas propriedades podemos alterar as características dos componentes.

A Figura 20 ilustra parcialmente o código-fonte em *Python* da interface gerado pelo *wxFormBuilder*.

Figura 20: Código gerado pelo wxFormBuilder.

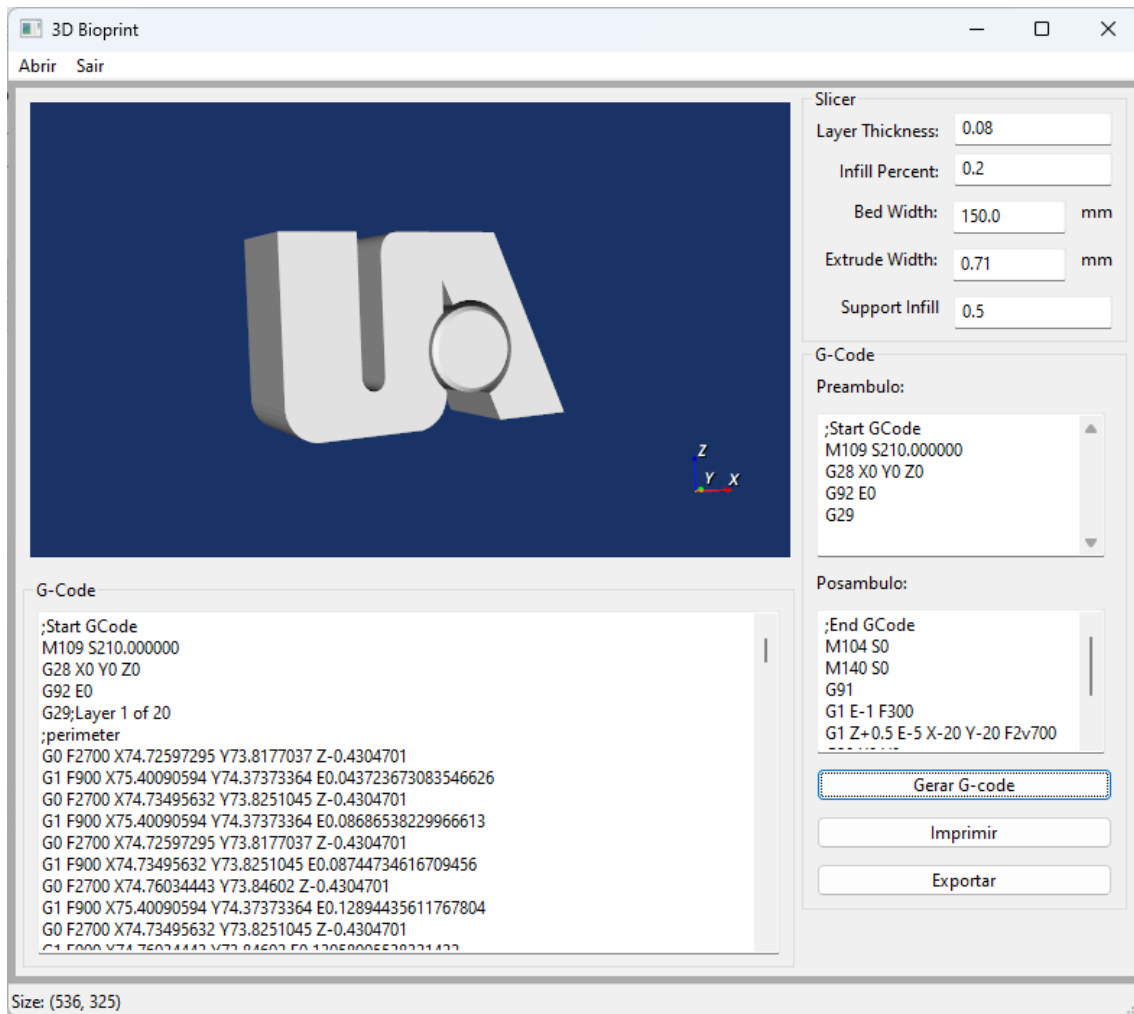


```
1  -*- coding: utf-8 -*-
2  #####
3  ## Python code generated with wxFormBuilder (version 3.10.1-0-g8feb16b3)
4  ## http://www.wxformbuilder.org/
5  ##
6  ## PLEASE DO *NOT* EDIT THIS FILE!
7  #####
8  import wx
9  import wx.xrc
10 ID_STL = 1000
11 #####
12 ## Class FrPrincipal
13 #####
14 class FrPrincipal ( wx.Frame ):
15     def __init__( self, parent ):
16         wx.Frame.__init__( self, parent, id = wx.ID_ANY, title = u"3D Bioprint",
17                             pos = wx.DefaultPosition,
18                             size = wx.Size( 715,700 ),
19                             style = wx.DEFAULT_FRAME_STYLE|wx.TAB_TRAVERSAL )
20         self.SetSizeHints( wx.DefaultSize, wx.DefaultSize )
21         bSizer2 = wx.BoxSizer( wx.VERTICAL )
22         bSizer2.SetMinSize( wx.Size( 800,800 ) )
23         self.m_panel1 = wx.Panel( self, wx.ID_ANY,
24                                   wx.DefaultPosition,
25                                   wx.DefaultSize,
26                                   wx.TAB_TRAVERSAL )
27         bSizer6 = wx.BoxSizer( wx.VERTICAL )
28         bSizer8 = wx.BoxSizer( wx.HORIZONTAL )
29         bSizer10 = wx.BoxSizer( wx.VERTICAL )
30         self.m_panel3 = wx.Panel( self.m_panel1, wx.ID_ANY,
31                                   wx.DefaultPosition
```

Fonte: Produzida pelo autor.

A primeira versão da interface do *BioSlicer* foi idealizada para ser simples e minimalista integrando as configurações de fatiamento e permitindo a edição rápida do *gCode* quando necessário. Ao abrir o *BioSlicer* o usuário pode visualizar o modelo 3D a ser impresso e ao lado do modelo 3D estão os parâmetros de fatiamento, o cabeçalho e rodapé do *gCode* e três botões. O primeiro para fatiar o modelo 3D e gerar o *gCode*, o segundo para enviar o código para a impressora e o terceiro para exportar o *gCode* para um arquivo texto. A primeira versão da interface do *plugin* pode ser vista na Figura 21.

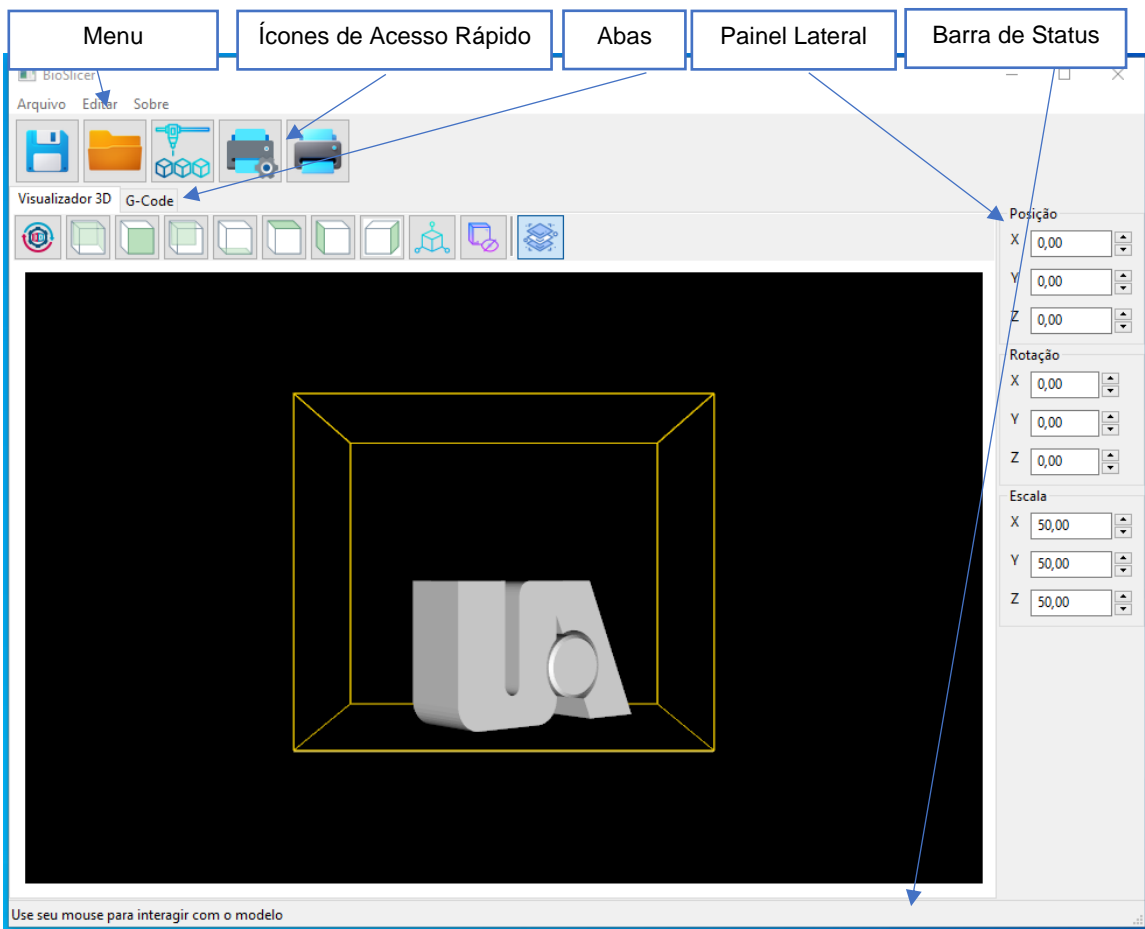
Figura 21: Primeira Versão da interface do Plugin.



Fonte: Produzida pelo autor.

A primeira versão do *plugin* foi atualizada e novos recursos foram adicionados. A versão atual da interface do *plugin BioSlicer* é composta por uma barra de menu principal com as opções Arquivo, Editar e Sobre. Uma barra de ícones de acesso rápido com atalhos para as seguintes funções: Salvar STL, Abrir arquivo STL, Parâmetros de Impressão, Configurar Impressora e Imprimir. Duas abas, uma para a visualização 3D do arquivo STL e ferramentas de visualização e edição simplificada e uma para a visualização do *gCode*. Um painel lateral com as informações sobre a posição, rotação e escala do objeto 3D nos eixos X,Y e Z. Uma barra de *Status*. A Figura 22 ilustra a nova versão da interface.

Figura 22: Versão atual da Interface do BioSlicer.



Fonte: Produzida pelo autor.

Além do *frame* principal outros três *frames* foram adicionados ao *plugin* durante a atualização. Um *frame* que contém as opções de configurações de impressão que é exibido ao clicar no menu Editar->Configuração de Impressão ou através do ícone de acesso rápido. A Figura 23 ilustra o frame Configuração de Impressão. Um *frame* que contém as opções de Configuração de Impressora que é exibido ao clicar no menu Editar->Configuração da Impressora ou através do ícone de acesso rápido. A Figura 24 ilustra o frame Configuração de Impressora. E um *frame* com informações do andamento da impressão que é exibido ao clicar no menu Arquivo->Imprimir ou através do ícone de acesso rápido. A Figura 25 ilustra o frame com informações do andamento da impressão.

Figura 23: Frame - Configuração de Impressão.

Configuração de Impressão

Qualidade | Suporte | Adesão | Retração | Materiais

Altura da Camada: 0,20 Altura da camada de corte em mm. (Min: 0.01, Max: 0.5, Padrão: 0.2)

Camadas Externas: 2 Número de paredes(cascas) externas para imprimir. (Min: 1, Max: 10, Padrão: 2)

Inícios Aleatórios: Habilitar randomização de inícios de perímetro. (Padrão: True)

Camadas Superiores: 0 Número de camadas a serem impressas na parte superior do objeto. (Min: 0, Max: 10, Padrão: 3)

Camadas Inferiores: 0 Número de camadas a serem impressas na parte inferior do objeto. (Min: 0, Max: 10, Padrão: 3)

Tipo de Preenchimento: Grid Padrão como o preenchimento será impresso. (Padrão: Grid)

Densidade de Preenchimento: 20 Densidade de preenchimento em porcentagem. (Min: 0, Max: 100, Padrão: 30)

Sobreposição de Preenchimento: 0,15 Quantidade, em mm, que o preenchimento se sobrepõe às extrusões do perímetro. (Min: 0.0, Max: 1.0, Padrão: 0.15)

Taxa de Alimentação: 10 Velocidade de extrusão (mm/s). (Min: 1, Max: 300, Padrão: 60)

Taxa de Deslocamento(xy): 150 Velocidade do movimento de deslocamento no plano xy (mm/s). (Min: 1, Max: 300, Padrão: 100)

Taxa de Deslocamento(z): 5,0 Velocidade de movimento do eixo Z (mm/s). (Min: 0.1, Max: 30, Padrão: 5)

Importar | Exportar | Restaurar Padrão | Salvar | Cancelar

Fonte: Produzida pelo autor.

Figura 24: Frame - Configuração da Impressora.

Configuração da Impressora

Velocidade da Porta Serial: 250000 Taxa de comunicação da porta serial da Impressora (Baud Rate). (Padrão: 115200)

Geometria da Mesa: Rectangular A forma da seção transversal do volume de construção. (Padrão: Rectangular)

Tamanho da Mesa (X): 200,0 O tamanho do eixo X da base da plataforma de construção. (Min: 0, Max: 1000, Padrão: 200)

Tamanho da Mesa (Y): 200,0 O tamanho do eixo Y da base da plataforma de construção. (Min: 0, Max: 1000, Padrão: 200)

Altura Eixo (Z): 170,0 Altura do eixo Z. (Min: 0, Max: 1000, Padrão: 200)

Centro da Mesa (X): 100,0 A coordenada X do centro da plataforma de construção. (Min: -500, Max: 500, Padrão: 100)

Centro da Mesa (Y): 100,0 A coordenada Y do centro da plataforma de construção. (Min: -500, Max: 500, Padrão: 100)

Temperatura da Mesa: 60 A temperatura para definir a plataforma de construção aquecida. (Min: 0, Max: 150, Padrão: 70)

Número de Extrusora: 1 O número de extrusoras que a impressora possui. (Min: 1, Max: 4, Padrão: 1)

Bico Padrão: 0 A extrusora padrão usada para impressão. (Min: 0, Max: 4, Padrão: 0)

Bico de Enchimento: -1 A extrusora usada para material de preenchimento. -1 significa usar bico padrão. (Min: -1, Max: 4, Padrão: -1)

Bico de Suporte: -1 A extrusora usada para material de suporte. -1 significa usar bico padrão. (Min: -1, Max: 4, Padrão: -1)

Bico 0

Temperatura do Bico: 150 A temperatura do bico para extrusora 0 (C). (Min: 150, Max: 250, Padrão: 190)

Diâmetro Filamento: 1,75 O diâmetro do filamento para extrusora 0 (mm). (Min: 1.0, Max: 3.5, Padrão: 1.75)

Diâmetro do Bico: 0,4 O diâmetro do bico para extrusora 0 (mm). (Min: 0.1, Max: 1.5, Padrão: 0.4)

Deslocamento em X: 0,0 O deslocamento posicional X para a extrusora 0 (mm). (Min: -100, Max: 100, Padrão: 0.0)

Deslocamento em Y: 0,0 O deslocamento posicional Y para a extrusora 0 (mm). (Min: -100, Max: 100, Padrão: 0.0)

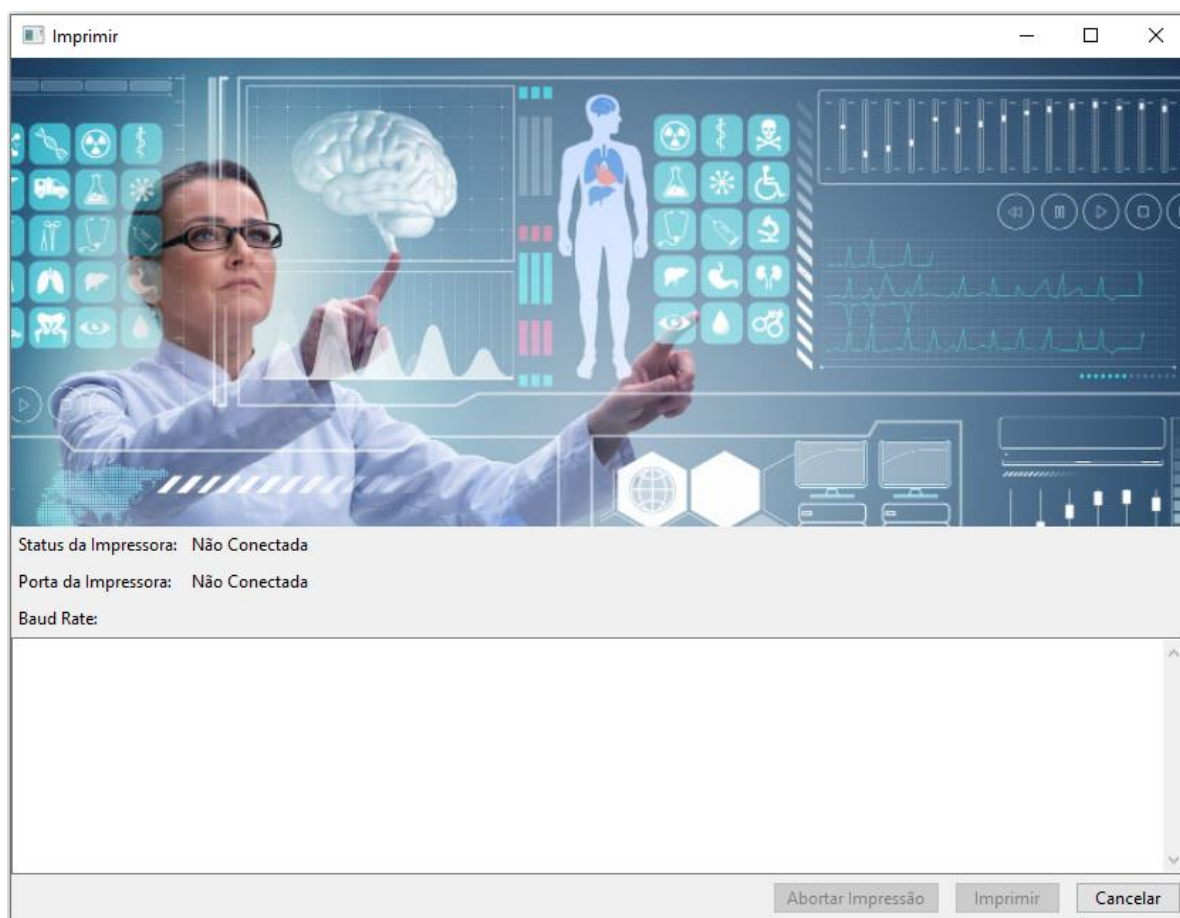
Velocidade Máxima: 50,0 A velocidade máxima ao usar a extrusora 0 (mm/s). (Min: 0, Max: 200, Padrão: 75.0)

Material: PLA Selecione o material a ser usado neste bico

Importar | Exportar | Restaurar Padrão | Salvar | Cancelar

Fonte: Produzida pelo autor.

Figura 25: Frame - Imprimir.



Fonte: Produzida pelo autor.

5.2 Features

A seguir serão apresentados os detalhes técnicos das funcionalidades (*features*) implementadas no *plugin*. São elas: Visualização do modelo, parametrização, fatiamento, *gCode*, Exportação e Impressão. A divisão das *features* ajuda a entender melhor como cada uma das funcionalidades funciona internamente no *plugin*.

5.2.1 Visualização do modelo 3D

O primeiro passo para que as funcionalidades do *plugin* sejam viabilizadas é o carregamento do arquivo no formato STL. Para compatibilizar o *plugin* com ambas as codificações (binária e ASCII) ao abrir o arquivo selecionado pelo

usuário (podendo ser binário ou ASCII), o sistema cria uma pasta temporária e utiliza a função *mesh.Mesh.from_file* do módulo *numpy-stl* para criar dois arquivos na pasta temporária, um em binário e outro em ASCII. Assim, o sistema utiliza a codificação necessária dependendo da necessidade da biblioteca.

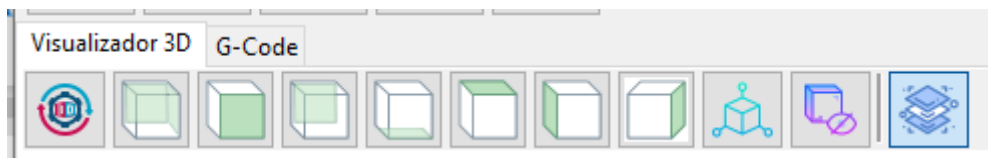
Para visualizar os modelos em 3D foi usada a biblioteca VTK (*Visualization Toolkit*) para *Python*. O VTK é um sistema de *software* de código aberto e disponível gratuitamente para computação gráfica 3D, modelagem, processamento de imagem, renderização de volume, visualização científica e plotagem 2D (Vtk, 2023).

A classe *vtkPanel* recebe como parâmetro um componente *wx.Panel*. A propriedade referente ao tamanho do Painel é definida para ser herdada da classe do *Painel* criada na interface. Um objeto *wxVTKRenderWindowInteractor* é criado para permitir a interação do usuário utilizando o mouse. O método *vtkRenderer* da classe VTK é chamado para fornecer uma especificação abstrata para renderizadores. Um renderizador é um objeto que controla o processo de renderização de objetos. A renderização é o processo de conversão de geometria, uma especificação para luzes e uma visão de câmera em uma imagem. O *vtkRenderer* também executa a transformação de coordenadas entre coordenadas mundiais, coordenadas de visualização (o sistema de coordenadas de renderização de computação gráfica) e coordenadas de exibição (as coordenadas reais da tela no dispositivo de exibição) (Davis, 2023).

O método *renderthis* da classe *vtkPanel* usa o método *GetRenderWindow().AddRenderer(self.renderer)* do objeto *wxVTKRenderWindowInteractor* para criar um renderizador. O método *FileDialog* da classe *wx* é chamado para que uma caixa de diálogo seja aberta e permita que usuário escolha o arquivo com a extensão *stl* que ele deseja que seja renderizado no *vtkPanel*. Os arquivos temporários são criados usando o método *createFiles*. O método *vtkSTLReader* da classe *vtk* é chamado para renderizar o arquivo *stl*. O método *vtkPolyDataMapper* é chamado para mapear os dados poligonais (ou seja, *vtkPolyData*) para gráficos primitivos. Um ator (objeto dentro da cena – imagem que será renderizada) é criado pelo método *vtkActor*. Um ícone para representar os eixos 3D é adicionado à cena através do método *vtkAxesActor*. Por fim os parâmetros da câmera são ajustados.

O módulo de visualização 3D possui algumas ferramentas que facilitam a visualização do objeto 3D pelo usuário, bem como sua manipulação. Na aba Visualizador 3D do *BioSlicer* está disponível uma barra de ferramenta com onze botões. A Figura 26 ilustra a barra de ferramenta do visualizador 3D.

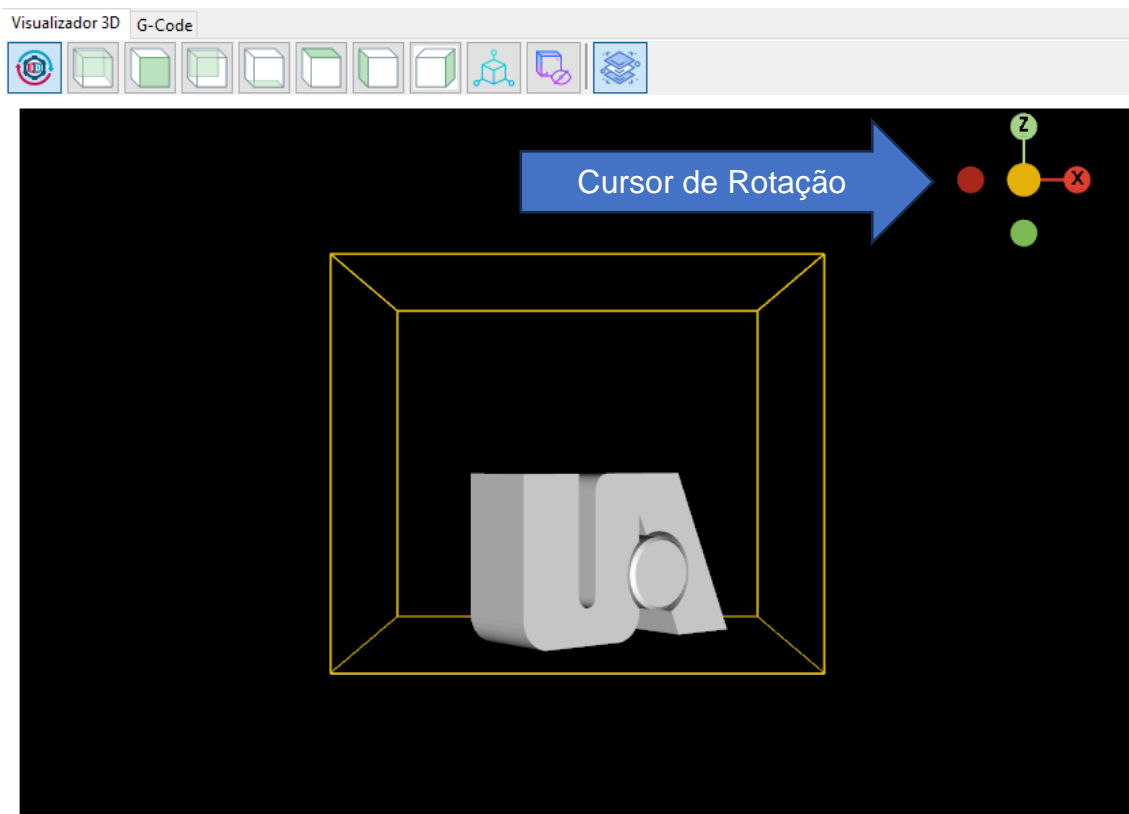
Figura 26: Barra de ferramentas do visualizador 3D.



Fonte: Produzida pelo autor.

O primeiro botão da barra de ferramenta é o Cursor de Rotação, ao clicar nesse botão um painel no canto superior direito do objeto 3D com a indicação dos eixos X, Y e Z é habilitado. O Cursor de Rotação permite rotacionar o objeto 3D no centro do seu eixo. A Figura 27 ilustra a visualização de uma imagem em 3D com o Cursor de Rotação Habilitado.

Figura 27: Visualizador 3D com Cursor de Rotação Habilitado.



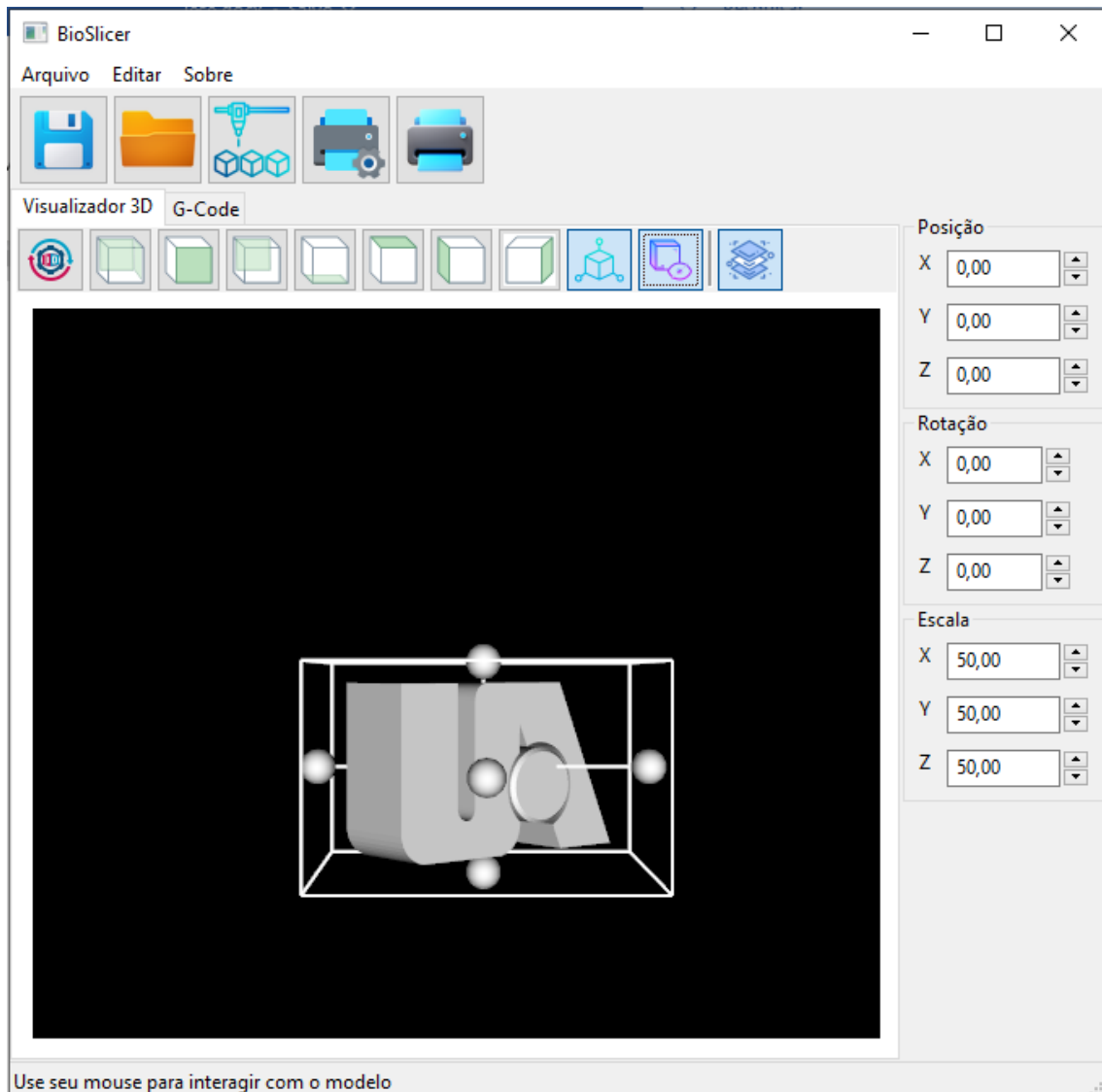
Fonte: Produzida pelo autor.

O segundo botão e os subsequentes até o oitavo são usados para posicionamento rápido da câmera para visualizações específicas do objeto. As opções de visualizações rápidas disponíveis seguindo a ordem dos botões na barra são:

- Isométrica: representa graficamente as três dimensões de um objeto em um único plano. É baseado num sistema de três semirretas que têm o mesmo ponto de origem e formam entre si três ângulos de 120°;
- Vista de Frente: mostra a projeção frontal do objeto;
- Vista Posterior: mostra o objeto sendo visto por trás;
- Vista Inferior: mostra o objeto sendo visto pelo lado de baixo;
- Vista Superior: mostra a projeção do objeto visto por cima;
- Vista Lateral Esquerda: mostra o objeto visto pelo lado esquerdo;
- Vista Lateral Direita: mostra o objeto visto pelo lado direito.

O nono botão da barra de ferramentas habilita a função editar. Esta função quando habilitada permite ao usuário, através do mouse, rotacionar o objeto 3D, alterar a escala do objeto 3D em qualquer eixo e mover o objeto 3D nos eixos X, Y e Z. A Figura 28 ilustra o objeto 3D com a opção editar habilitada.

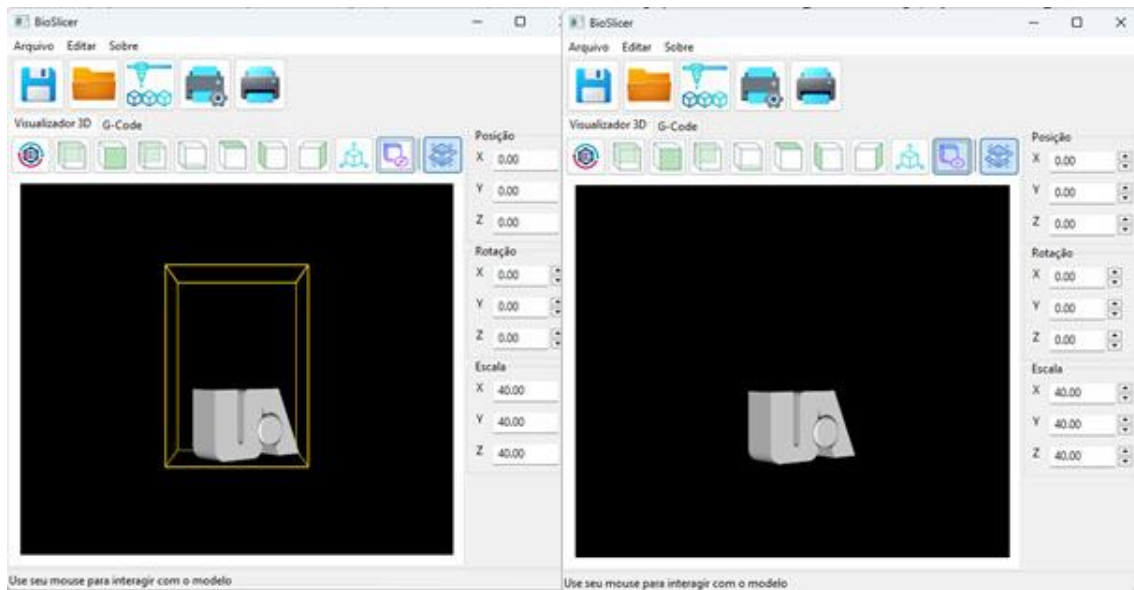
Figura 28: Objeto 3D com a opção editar habilitada.



Fonte: Produzida pelo autor.

O décimo botão da barra de ferramentas habilita/desabilita a opção ver/ocultar linhas da impressora que representam os limites da impressora. Esta função é útil para que o usuário visualize se o objeto 3D está ultrapassando o limite da impressora. Figura 29 ilustra um objeto 3D com a opção Ver/Ocultar Linhas da Impressora, no lado direito a opção está habilitada e no lado esquerdo a opção está desabilitada.

Figura 29: Função Ver/Ocultar Limites da Impressora.



Fonte: Produzida pelo autor.

O décimo primeiro botão da barra é o botão fatiar. Ao clicar nesse botão o *gCode* será gerado a partir do objeto 3D.

5.2.2 Parametrização

O *plugin* possui duas seções para que o usuário parametrize o sistema antes da impressão. Uma para as configurações de impressão e outra para as configurações da impressora.

5.2.2.1 Configurações de Impressão

Os parâmetros de impressão definem como o material será depositado camada a camada. O algoritmo de fatiamento cria cada camada de acordo com os parâmetros definidos. A configuração dos parâmetros de impressão está dividida em cinco categorias: Qualidade, Suporte, Adesão, Retração e Materiais. A seguir será descrito os parâmetros do sistema agrupados por categoria.

- Qualidade:
 - Altura da Camada: Altura da camada de corte em mm. (Mín: 0.01, Máx: 0.5, Padrão: 0.2);

- Camadas Externas: Número de paredes(cascas) externas para imprimir. (Mín: 1, Máx: 10, Padrão: 2);
- Inícios Aleatórios: Habilitar randomização de inícios de perímetro. (Padrão: True);
- Camadas Superiores: Número de camadas a serem impressas na parte superior do objeto. (Mín: 0, Máx: 10, Padrão: 3);
- Camadas Inferiores: Número de camadas a serem impressas na parte inferior do objeto. (Mín: 0, Máx: 10, Padrão: 3);
- Tipo de Preenchimento: Padrão como o preenchimento será impresso. (Padrão: Grid);
- Densidade de Preenchimento: Densidade de preenchimento em porcentagem. (Mín: 0, Máx: 100, Padrão: 30);
- Sobreposição de Preenchimento: Quantidade, em mm, que o preenchimento se sobrepõe às extrusões do perímetro. (Mín: 0.0, Máx: 1.0, Padrão: 0.15);
- Taxa de Alimentação: Velocidade de extrusão (mm/s). (Mín: 1, Máx: 300, Padrão: 60);
- Taxa de Deslocamento(xy): Velocidade do movimento de deslocamento no plano xy (mm/s). (Mín: 1, Máx: 300, Padrão: 100);
- Taxa de Deslocamento(z): Velocidade de movimento do eixo Z (mm/s). (Mín: 0.1, Máx: 30, Padrão: 5).
- Suporte:
 - Tipo de Suporte: Que tipo de estrutura de suporte adicionar. (Padrão: External);
 - Início do Suporte: A que distância as estruturas de suporte devem ser impressas do modelo, horizontalmente. (Mín: 0, Máx: 2, Padrão: 0.5);
 - Densidade de Suporte: Densidade interna da estrutura de suporte. (Mín: 0, Máx: 100, Padrão: 33.0);
 - Ângulo de Ressalto: Ângulo da vertical para o qual as estruturas de suporte devem ser impressas. (Mín: 0, Máx: 90, Padrão: 45).
- Adesão:
 - Tipo de Adesão: O tipo de estrutura de adesão da base. (Padrão: None);

- Largura da Aba: Largura da aba para imprimir na primeira camada para ajudar na adesão da peça. (Mín: 0, Máx: 20, Padrão: 3.0);
- Camadas da Plataforma: Número de camadas a serem usadas na fabricação da plataforma. (Mín: 1, Máx: 5, Padrão: 3);
- Topo da Plataforma: Quão alta a plataforma estará em relação à mesa. (Mín: 0, Máx: 50, Padrão: 3.0);
- Início da Linha Limite: Quão longe a linha limite deve ser impressa do modelo. (Mín: 0, Máx: 20, Padrão: 0.0);
- Camadas da Linha Limite: Número de camadas impressas na linha limite. (Mín: 0, Máx: 1000, Padrão: 0);
- Comprimento Principal: Comprimento do filamento a ser extrudado ao preparar hotends. (Mín: 0, Máx: 1000, Padrão: 10.0).
- Retração:
 - Habilitar Retração: Habilita a retração do filamento. (Padrão: True);
 - Velocidade de Retração: Velocidade para retrain o filamento em (mm/s). (Mín: 0, Máx: 200, Padrão: 30.0);
 - Distância de Retração: Distância para retrain o filamento entre os movimentos de extrusão (mm). (Mín: 0, Máx: 20, Padrão: 3.0);
 - Distância de Retração (extrusora): Distância para retrain o filamento na troca da extrusora (mm). (Mín: 0, Máx: 50, Padrão: 3.0);
 - Levantar Cabeça Extrusora: Distância para levantar a cabeça extrusora durante movimentos de retração (mm). (Mín: 0, Máx: 10, Padrão: 0.0).
- Materiais:
 - Escolha o tipo de Material: Define o tipo de material que está sendo utilizado para a impressão;
 - Temperatura da Cama: A temperatura do leito a ser usada para material selecionado;
 - Temperatura da Extrusora: A temperatura da extrusora a ser usada para material selecionado;
 - Velocidade Máxima: A velocidade máxima ao extrudar o material selecionado.

5.2.2.2 Configurações da Impressora

As configurações da impressora permitem ao usuário ajustar os parâmetros de impressão para uma determinada impressora 3D ou bioimpressora 3D. O *plugin* possui os seguintes parâmetros de Impressora:

- Velocidade da Porta Serial: Taxa de comunicação da porta serial da Impressora (Baud Rate). (Padrão: 115200);
- Geometria da Mesa: A forma da seção transversal do volume de construção. (Padrão: Rectangular);
- Tamanho da Mesa(X): O tamanho do eixo X da base da plataforma de construção. (Mín: 0, Máx: 1000, Padrão: 200);
- Tamanho da Mesa (Y): O tamanho do eixo Y da base da plataforma de construção. (Mín: 0, Máx: 1000, Padrão: 200);
- Altura Eixo (Z): Altura do eixo Z. (Mín: 0, Máx: 1000, Padrão: 200);
- Centro da Mesa(X): A coordenada X do centro da plataforma de construção. (Mín: -500, Máx: 500, Padrão: 100);
- Centro da Mesa (Y): A coordenada Y do centro da plataforma de construção. (Mín: -500, Máx: 500, Padrão: 100);
- Temperatura da Mesa: A temperatura para definir a plataforma de construção aquecida. (Mín: 0, Máx: 150, Padrão: 70);
- Número de Extrusora: O número de extrusoras que a impressora possui. (Mín: 1, Máx: 4, Padrão: 1);
- Bico Padrão: A extrusora padrão usada para impressão. (Mín: 0, Máx: 4, Padrão: 0);
- Bico de Enchimento: A extrusora usada para material de preenchimento. -1 significa usar bico padrão. (Mín: -1, Máx: 4, Padrão: -1);
- Bico de Suporte: A extrusora usada para material de suporte. -1 significa usar bico padrão. (Mín: -1, Máx: 4, Padrão: -1).

O *plugin* permite trabalhar com até quatro bicos de extrusão. Para cada bico de extrusão é possível configurar os seguintes parâmetros:

- Temperatura do Bico: A temperatura do bico para extrusora 0 (C). (Mín: 150, Máx: 250, Padrão: 190);

- Diâmetro Filamento: O diâmetro do filamento para extrusora 0 (mm). (Mín: 1.0, Máx: 3.5, Padrão: 1.75);
- Diâmetro do Bico: O diâmetro do bico para extrusora 0 (mm). (Mín: 0.1, Máx: 1.5, Padrão: 0.4);
- Deslocamento em X: O deslocamento posicional X para a extrusora 0 (mm). (Mín: -100, Máx: 100, Padrão: 0.0);
- Deslocamento em Y: O deslocamento posicional Y para a extrusora 0 (mm). (Mín: -100, Máx: 100, Padrão: 0.0);
- Velocidade Máxima: A velocidade máxima ao usar a extrusora 0 (mm/s). (Mín: 0, Máx: 200, Padrão: 75.0);
- Material: define o material a ser usado no bico selecionado.

5.2.3 Algoritmo de Fatiamento (*Slicer*)

Na MA, a separação em camadas que formarão o volume do objeto a ser impresso é conhecida como fatiamento. Esta é a etapa mais importante no planejamento do processo da MA (Livesu *et al.*, 2017), pois sem ela não é possível realizar a impressão 3D. Segundo Volpato (2007), a obtenção dos contornos é feita através da intersecção do modelo 3D com camadas em diferentes alturas. Os algoritmos de fatiamento buscam encontrar os contornos através do resultado da intersecção do modelo com as camadas (Livesu *et al.*, 2017).

Basicamente o algoritmo de fatiamento em malhas triangulares identifica os seguimentos de retas oriundos das intersecções dos triângulos da malha com as camadas, juntando os pontos finais das retas formando o contorno do objeto (Kirschman e Jaraalmonite, 1992

O padrão STL não contempla informações sobre triângulos adjacentes o que obriga o algoritmo de fatiamento a definir a sequência e direção das arestas para que se possa uni-las adequadamente, a fim de formar o contorno da camada (Kirschman e Jara-Almonite, 1992). O processo de unir as arestas pode ser feito procurando por pontos que estejam sobre o mesmo local ou classificando os pontos com menor distância entre eles (Volpato, 2007).

O arquivo STL no padrão ASCII deve começar com a linha *solid nome*, onde *nome* é uma cadeia qualquer de caracteres. Após a primeira linha, cada triângulo é definido iniciando na instrução *facet normal nx ny nz*, onde *nx ny nz* são valores em notação ponto flutuante que representam as coordenadas do vetor normal do triângulo, e encerrando na instrução *endfacet*. Encapsulados pela instrução *facet* estão os vértices dos triângulos indicados pela instrução *vertex* seguidos das coordenadas em ponto flutuante *x,y,z*. A Figura 30 ilustra o formato de arquivos STL em ASCII.

Figura 30: Formato de arquivos STL em ASCII.

```
1  solid nome
2  facet normal nx ny nz
3      outer loop
4          vertex v1x v1y v1z
5          vertex v2x v2y v2z
6          vertex v3x v3y v3z
7      endloop
8  endfacet
9  endsolid nome
```

Fonte: Produzida pelo autor.

O algoritmo de fatiamento utilizado neste trabalho foi adaptado do algoritmo desenvolvido por Revar Desmera, e está disponível gratuitamente em <https://github.com/revarbat/mandoline-py?tab=BSD-2-Clause-1-ov-file#readme>. O algoritmo foi escrito em *python* e utiliza as bibliotecas *math*, *sys*, *string* e *copy* para percorrer os vértices do arquivo STL, fatiar o modelo e gerar o código G para então ser enviado para a impressora.

5.2.4 Código *gCode* gerado pelo algoritmo

O *gCode* gerado durante o fatiamento a partir do modelo 3D será interpretado pela impressora 3D. Cada impressora possui um *software* embarcado denominado de *firmware*. O *firmware* é responsável por gerenciar todas as operações realizadas pelo *hardware* da impressora 3D. Ele coordena os aquecedores, sensores, luzes, display LCD, botões e tudo o mais relacionado ao processo de impressão 3D (Marlin, 2023).

Existem diversos *firmwares* disponíveis no mercado. A seguir, serão apresentados os principais *firmwares* de código aberto:

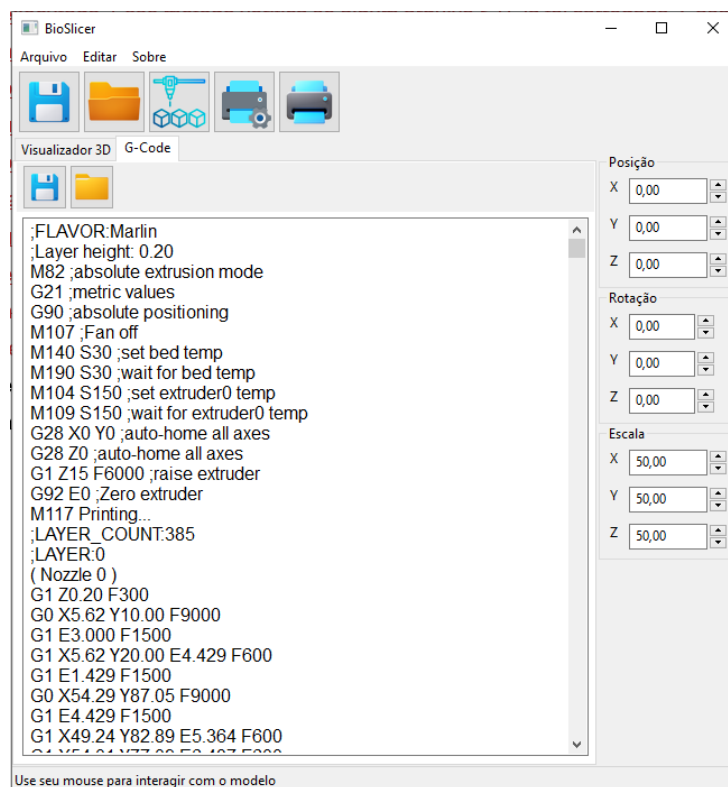
- **Sprinter:** compatível com *hardware* RAMPS (*RepRap Arduino Mega Pololu Shield*) e outras configurações eletrônicas de processador único *RepRap*, suporta impressão a partir de cartão SD e controle de leito térmico ativo. Seu código-fonte está licenciado sob a GNU GPL v3 ou (a critério do usuário) qualquer versão posterior. É baseado no firmware da Tonokips, que foi licenciado sob GPL v2 ou posterior (Reprap, 2023a);
- **Marlin:** é compatível com a família *RepRap* de prototipadores rápidos de replicação. Foi derivado do *Sprinter* e do *grbl* e tornou-se um projeto autônomo de código aberto em 12 de agosto de 2011 com seu lançamento no *Github*. O *Marlin* é licenciado sob a GPLv3 e é gratuito para todos os aplicativos. É usado por várias impressoras 3D comerciais, tais como: *LulzBot*, *Průša Research*, *Creality3D*, *BIQU*, *Geeetech* e *Ultimaker*. Também é capaz de conduzir máquinas CNC e gravadores a laser (Marlin, 2023);
- **Repetier:** compatível com placas *RepRap* baseadas em *Arduino* com pelo menos 64kB de memória *flash*. Foi escrito a partir do *firmware Sprinter*, possui suporte para *multiextrusor*, *dry-run* (permite detectar falhas não relacionadas com o extrusor), cartão SD e painéis LCD. Também é possível adaptá-lo para impressão com mistura de cores (Repetier, 2023);
- **Teacup:** compatível com controladores *RepRap* suporta 4 eixos, número arbitrário de aquecedores, número arbitrário de sensores de temperatura e *multiextrusor* (Reprap, 2023b);
- **RepRap:** compatível com microprocessadores de 32 *bits*, possui recursos de calibração automática, suporte a rede via *Ethernet* ou *Wi-Fi* e cartão SD. Outro recurso é a capacidade de reportar o estado de toda a impressora através de um único comando de código G, o M408 (Reprap, 2023c).

O *gCode* é interpretado de maneira distinta por diferentes *firmwares*. Por exemplo, a instrução M85 é suportada pelos *firmwares Sprinter*, *Marlin* e

Repetier, mas não pelo *Teacup* ou o *RepRap* (Campos, 2019). Neste presente trabalho, o código gerado é compatível com o *firmware Marlin*, pois é o mais comum em bioimpressoras.

Assim que o botão Fatiar é acionado no *BioSlicer* o método *gerarGcode* é executado. Esse método verifica se algum arquivo STL já foi carregado. Caso nenhum arquivo tenha sido carregado a seguinte mensagem será exibida para o usuário: “Não existe nenhum modelo 3D para ser fatiado!”. Se já existir um modelo carregado em memória um arquivo temporário será criado com os dados existentes na memória e o arquivo é carregado pelo algoritmo de fatiamento através do método *read_file* da classe *StlData*. O código *gCode* gerado pelo algoritmo de fatiamento é então carregado no objeto *wxTextCtrl* e exibido para o usuário na aba *G-Code* do *BioSlicer*. O usuário pode editar o código *gCode* gerado no *BioSlicer* antes de enviar para a impressora. A Figura 31 ilustra a tela do *BioSlicer* com o *gCode* gerado.

Figura 31: gCode gerado no BioSlicer.



Fonte: Produzida pelo autor.

Por padrão o *plugin* carrega as seguintes instruções no início do *gCode*:

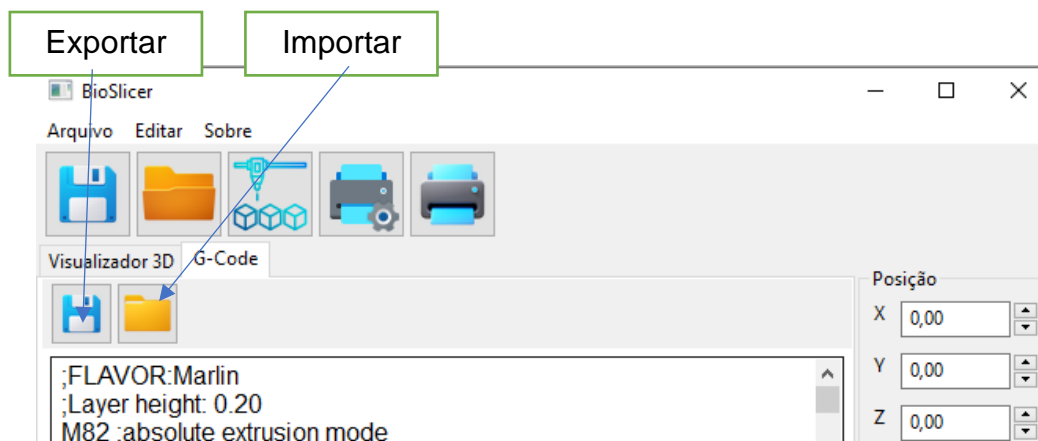
- M109 S210.000000: instrução para aquecimento da cabeça de impressão a 210°C;
- G28 X0 Y0 Z0: instrução para posicionar a cabeça de impressão nas coordenadas x=0, y=0 e z=0;
- G92 E0: define a nova posição da extrusora para o ponto 0.

. Em seguida, são definidos os comandos G0 e G1 para adicionar um movimento linear à fila a ser executada após a conclusão de todos os movimentos anteriores.

5.2.5 Exportar/Importar gCode

O *plugin* tem a função de importar um código *gCode* já existente de um arquivo, bem como, exportar o código *gCode* gerado no *plugin* para um arquivo com extensão *gCode*. Para exportar o código *gCode* gerado no *plugin* o método *exportarGcode* chama um objeto *wx.FileDialog* para abrir uma janela e o usuário definir o nome e o local onde o arquivo será gravado. Após as informações do arquivo serem definidas, o método *open* do *Python* é invocado para criar o arquivo e o método *file.write* é chamado recebendo como parâmetro um *textCtrl.GetValue()* para transferir os dados do objeto para o arquivo. As funções de exportar arquivo então disponíveis na Aba *G-Code* do *BioSlicer*. A Figura 32 mostra as opções de importação e exportação do código *gCode* no *plugin*.

Figura 32: Opções de Exportar/Importar gCode.



Fonte: Produzida pelo autor.

5.2.6 Impressão do modelo

Um dos objetivos do trabalho é fazer com que o usuário não precise usar vários *softwares* para imprimir o modelo 3D, mantendo o usuário na mesma plataforma. Algumas bioimpressoras possuem porta USB (*Universal Serial Bus*) para comunicação com os computadores. Para implementar a comunicação via porta USB, as APIs *serial.tools.list_ports* e *serial* foram usadas. O primeiro passo para enviar os dados para a impressora é detectar se existe alguma impressora conectada na porta USB; para isso é necessário analisar todas as portas e verificar se o dispositivo conectado é uma impressora. Após detectar a porta, é preciso estabelecer o *baudRate* que é a taxa na qual os *bits* são transferidos. É importante ressaltar que tanto o computador quanto o dispositivo periférico devem ser configurados para a mesma taxa de transmissão antes que seja feita a leitura ou gravação dos dados com sucesso. O *baudRate* das impressoras testadas no trabalho são: 115200 bps para a impressora Ender 3 Pro e 250000 bps para a bioimpressora Genesis.

Ao clicar no botão Imprimir do *BioSlicer* um arquivo temporário com o código *gCode* atualizado é gerado. O arquivo temporário é necessário, pois o usuário pode editar o código *gCode* na tela antes de enviar para a impressora. Após o arquivo temporário ser criado o Frame Imprimir é carregado. No construtor da Classe do Frame Imprimir os métodos *getBaudRate* e *detectPrinterPort* são chamados. O primeiro método carrega o valor do *BaudRate* selecionado pelo usuário nos parâmetros da Impressora e o armazena em uma variável de memória para ser usado como parâmetro na comunicação da porta serial. O segundo método escaneia as portas seriais do computador e verifica se existe alguma impressora conectada. Se nenhuma impressora for detectada, o botão imprimir é desabilitado e a mensagem “Não conectada” será apresentada na frente do *label status* da impressora, bem como, do *label* porta da impressora. Caso o método *detectPrinterPort* encontre um dispositivo conectado à frente do *label status* Impressora será apresentada a mensagem “Conectada” e na frente do *label* porta da impressora o nome da porta serial em que a impressora está conectada.

Além de mostrar se existe uma impressora conectada no computador o Frame Imprimir possui um campo texto que exibe a porcentagem do *gCode* que

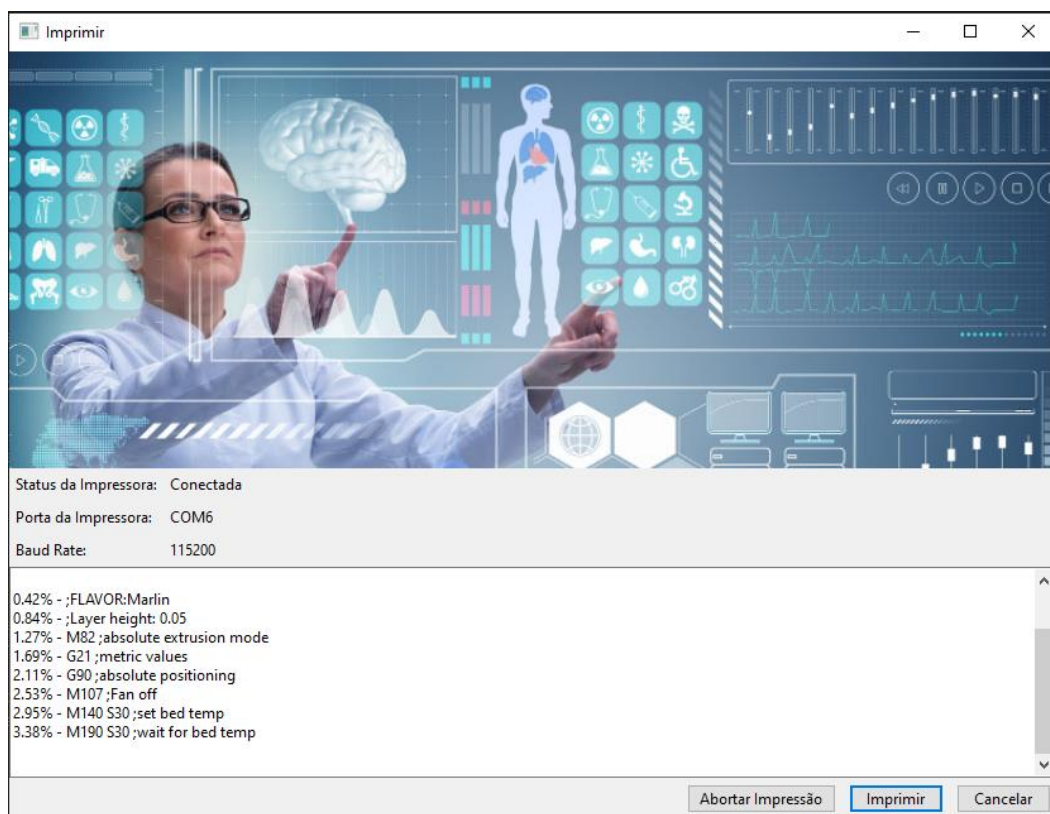
foi enviado para a impressora e o comando *gCode* que está sendo enviado, com esse recurso é possível acompanhar o progresso da impressão. A Figura 33 ilustra o Frame Imprimir com as informações de conexão da impressora e a progressão da impressão.

Uma vez que a impressora esteja conectada e o botão imprimir é pressionado uma *thread* é iniciada e o processo de envio do *gCode* via porta USB é iniciado. O uso da *thread* isola o processo de impressão impedindo o travamento da tela.

Em seguida, utilizou-se o comando *serial.write()* para enviar os comandos *gCodes* gerados na etapa anterior para a porta da impressora. O *gCode* deve ser enviado linha a linha e aguardar a resposta positiva (“ok”) da impressora lida através do comando *serial.readline()*.

Após todo o arquivo ser enviado para a impressora a conexão com a porta USB é encerrada com o comando *serial.close()*, e a impressora bem como o *plugin* estão prontos para repetir essa etapa se necessário.

Figura 33: Frame Imprimir com a exibição do progresso da impressão.



Fonte: Produzida pelo autor.

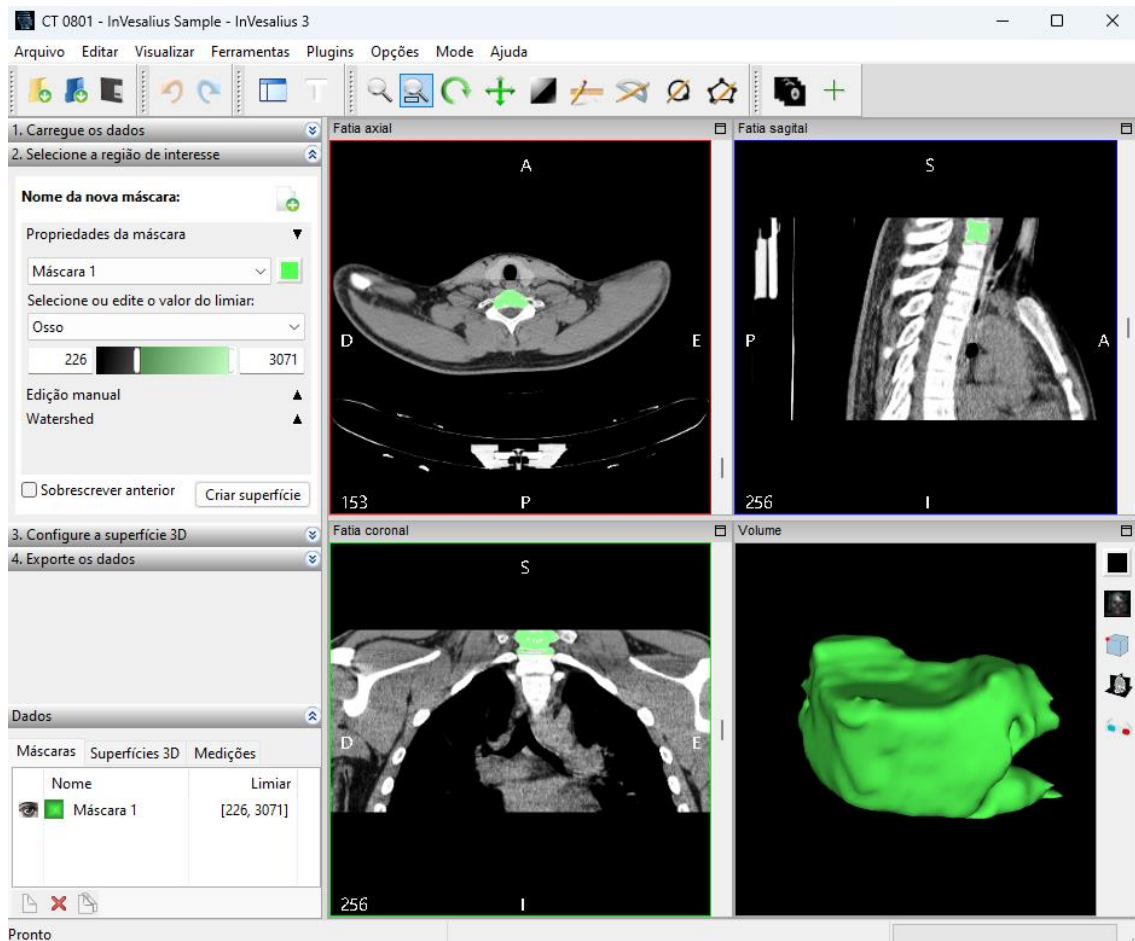
5.3 Validação do *plugin*

Para validar o *plugin* foram feitos testes de impressão de uma estrutura rígida em uma impressora 3D utilizando filamento e a impressão com hidrogel (alginato 2% + gelatina 6%) e creme hidratante (Loção Deo Hidratante comercial) em uma bioimpressora 3D.

5.3.1 Teste de Impressão com Filamento

Para o teste de impressão da estrutura rígida, foi importado no *InVesalius* um arquivo proveniente de imagens médicas no padrão DICOM. O arquivo foi obtido do site <https://invesalius.github.io/download.html> e está disponível para *download* na seção *DICOM IMAGES FOR TESTS* com o nome de *Clavicle*. Após a importação do arquivo DICOM no *InVesalius*, uma vértebra foi segmentada utilizando a ferramenta de corte. A Figura 34 ilustra a imagem segmentada. A ferramenta de corte foi acessada através do menu ferramentas -> máscara->corte. Após acionar a ferramenta de corte uma segmentação macro da vértebra foi feita nas fatias axial, sagital e coronal. Após o primeiro corte a ferramenta de segmentação manual foi usada para fazer pequenos ajustes na imagem segmentada.

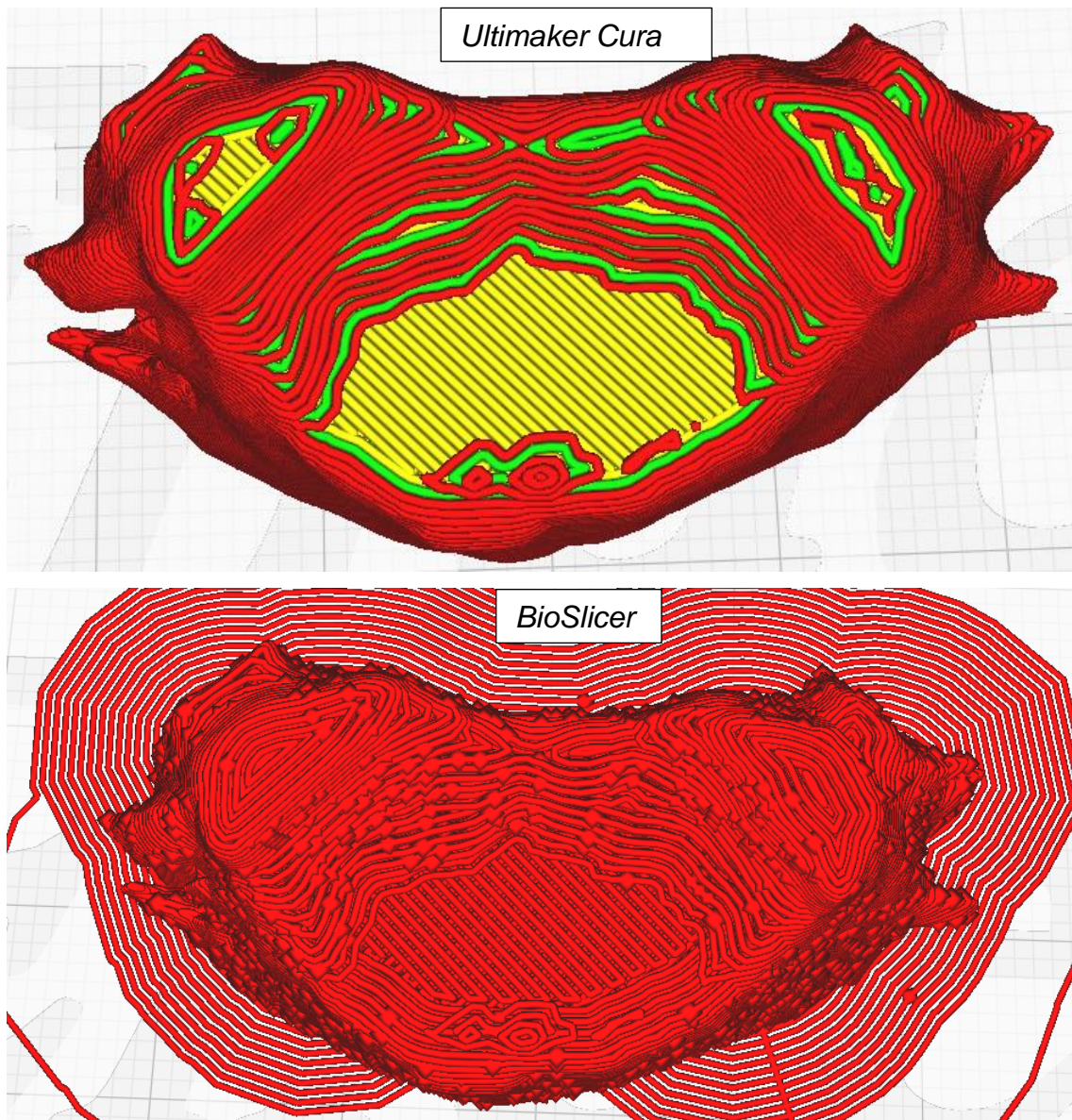
Figura 34: Segmentação de imagem para testes de impressão.



Fonte: Produzida pelo autor.

Depois de segmentar a imagem um arquivo STL foi gerado. Como forma de controle do teste de impressão em impressora 3D, o arquivo STL gerado pelo *InVesalius* foi fatiado pelo *plugin BioSlicer* e pelo *software Ultimaker Cura 5.2.1*. A Figura 35 mostra uma comparação entre o *gCode* gerado no *Ultimaker Cura 5.2.1* e no *BioSlicer*.

Figura 35: Comparação entre o gCode gerado pelo Ultimaker e o BioSlicer.



Fonte: Produzida pelo autor.

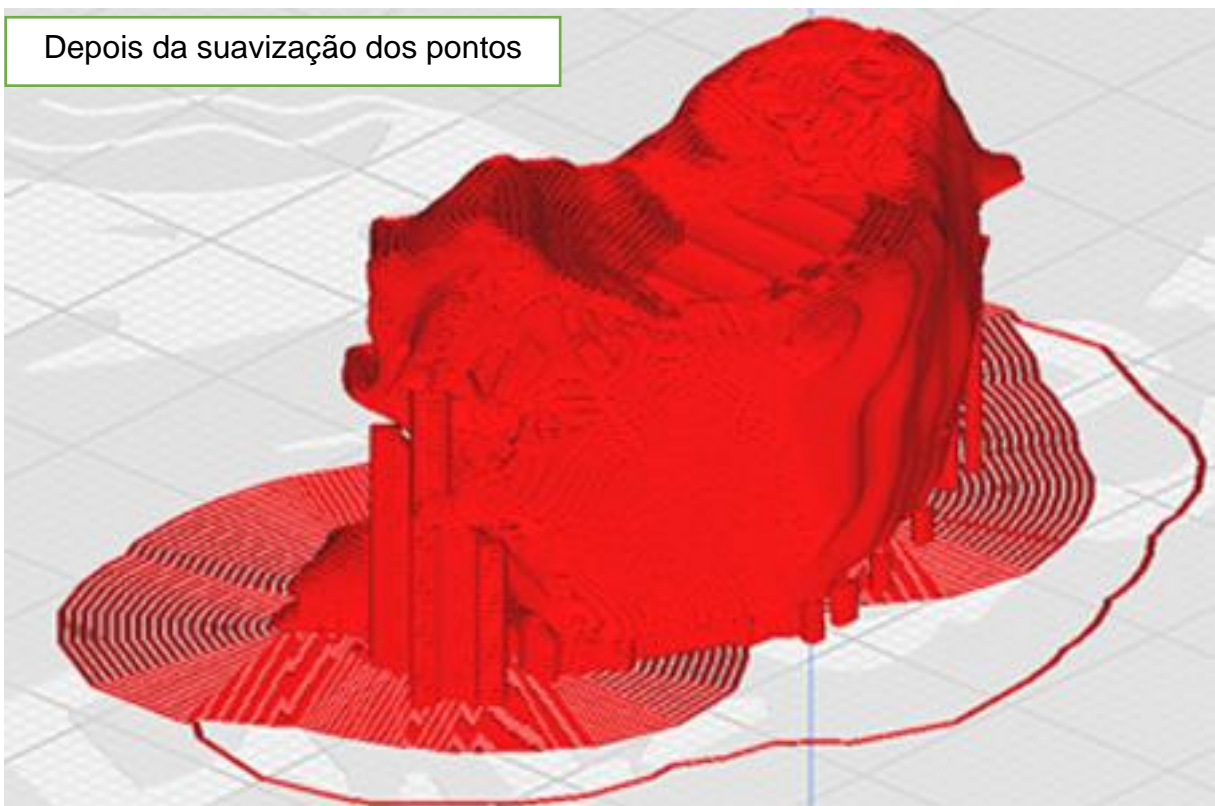
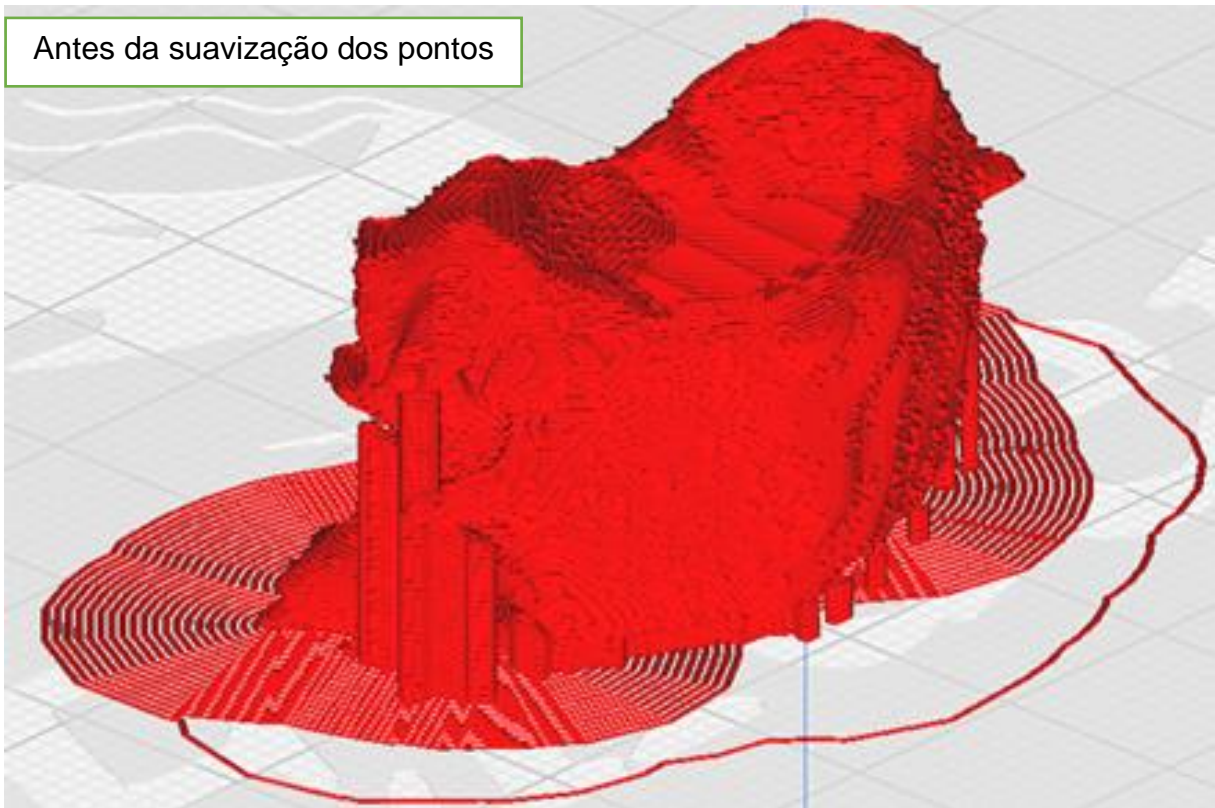
Pode-se notar algumas diferenças no gCode gerado pelos softwares. O *Ultimaker* marca as camadas por cores e permite visualizar o suporte separadamente, isso ocorre pois como está sendo usado o visualizador do *Ultimaker* para comparar as estruturas, o software *Ultimaker* identifica as diferentes regiões ao gerar o gCode. Já o gCode gerado pelo *BioSlicer* apresenta o *Brim*, o suporte junto com o restante da peça a ser impressa como um bloco único no visualizador. Pode-se notar que o gCode gerado pelo *BioSlicer* apresenta diversos pontos espalhados pelas linhas. Isto deve-se ao fato do

algoritmo de fatiamento usado no *Bioslicer* não eliminar pontos em X e Y próximos na mesma camada, criando assim um acúmulo de material em pontos específicos, formando pontos indesejáveis.

Para eliminar os pontos foi implementado um algoritmo de suavização de pontos que foi incorporado ao *BioSlicer*. Basicamente o algoritmo identifica no *gCode* pontos em X e Y com coordenadas muito próximas e as elimina. Para identificar os pontos que precisam ser eliminados o algoritmo verifica a posição de X e Y da linha atual e subtrai o ponto X e Y da próxima linha. Se a diferença estiver entre -0,04mm e +0,04mm, a linha atual é eliminada. Assim, na impressão, o material será depositado uma única vez no ponto (x, y).

A Figura 36 ilustra, na parte superior, a visualização do *gCode* sem a suavização dos pontos e, na parte inferior, com a suavização dos pontos.

Figura 36: Antes e depois de suavizar os pontos no gCode.



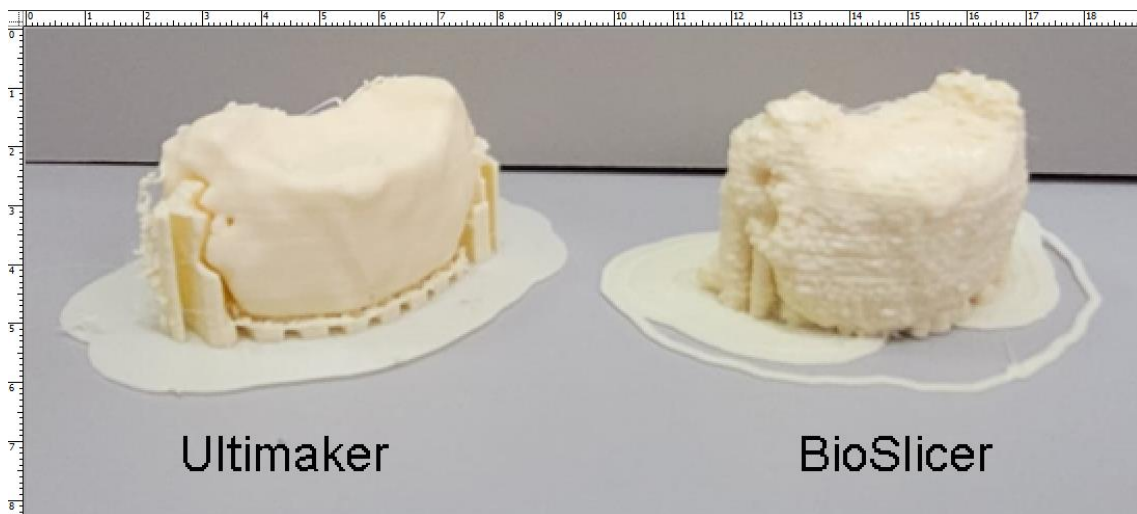
Fonte: Produzida pelo autor.

Após suavizar os pontos repetidos no *gCode* o modelo foi impresso. A impressora utilizada para a impressão em ambos os *softwares* foi a *Creativity Ender 3 Pro* com filamento PLA. Os parâmetros definidos para a impressão foram:

- Altura da camada: 0,2 mm
- Velocidade de Impressão: 50 mm/s
- Temperatura da Mesa: 60° C
- Temperatura do Bico: 205° C
- Espessura do Bico: 0,4 mm
- Diâmetro do Filamento: 1,75 mm

A Figura 37 mostra, à direita, a impressão feita pelo *software Ultimaker Cura* e, à esquerda, a impressão feita pelo *plugin BioSlicer*.

Figura 37: Comparação entre os modelos impressos no Ultimaker e no BioSlicer.



Fonte: Produzida pelo autor.

Ambos os modelos apresentaram largura, altura e profundidade iguais. O *Brim* do modelo impresso pelo *software Ultimaker* apresenta uniformidade ao redor de toda a peça. Já o *Brim* do modelo impresso pelo *software BioSlicer* não contorna a peça inteira, mas ambas as peças permaneceram grudadas na mesa durante a impressão dos modelos, mostrando a eficácia do *Brim* gerado em ambos os *softwares*. Ambos os *softwares* criam suporte, apesar da geometria

diferente no suporte gerado, ambos cumpriram a função de segurar o material da peça no local desejado. O acabamento da camada externa do modelo gerado pelo *Ultimaker* é mais liso, isso ocorre, pois o *Ultimaker* permite configurações diferentes como, por exemplo, altura da camada e velocidade de impressão, para a impressão das camadas externas, propiciando uma melhor finalização da peça.

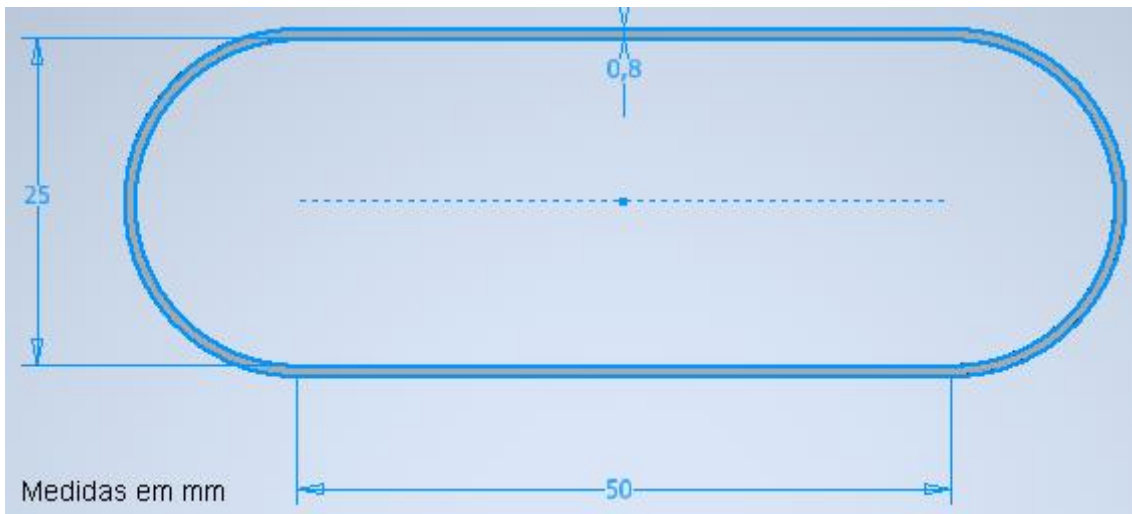
Apesar das diferenças apresentadas, o modelo impresso pelo *BioSlicer* manteve as características originais da estrutura segmentada no *InVesalius*, gerando um modelo físico que representa o modelo virtual.

5.3.2 Teste de Impressão com Creme Hidratante e Hidrogel

O objetivo do teste de impressão com uma bioimpressora 3D utilizando hidrogel e creme hidratante é verificar se a impressão mantém as características geométricas do *design* original. Optou-se em imprimir com creme hidratante pois é um produto facilmente encontrado no comércio e é viscoso suficiente para não colapsar a estrutura quando impresso em poucas camadas. O hidrogel possui viscosidade e características diferentes do creme hidratante.

Para realizar o teste, optou-se em criar um modelo no *software Autodesk Inventor* com dimensões pré-definidas e não um modelo gerado pelo *InVesalius*. A criação do modelo com medidas pré-definidas permite uma melhor aferição da geometria do material pós impressão. Outro motivo pelo qual optou-se por um modelo controlado é que, devido às características dos materiais não sólidos, dependendo do número de camadas a ser impressa, a estrutura pode colapsar. Como o objetivo do *BioSlicer* é a integração dos processos de fatiamento e impressão, bem como, ser usado como um *plugin* do *InVesalius*, o teste deve demonstrar a capacidade do *software* de fatiar o modelo e controlar a impressora, de tal forma que a impressão final siga a geometria original. A Figura 38 ilustra o modelo gerado no *software Autodesk Inventor* para a realização dos testes com a bioimpressora 3D.

Figura 38: Modelo criado no Autodesk Inventor para teste na bioimpressora.



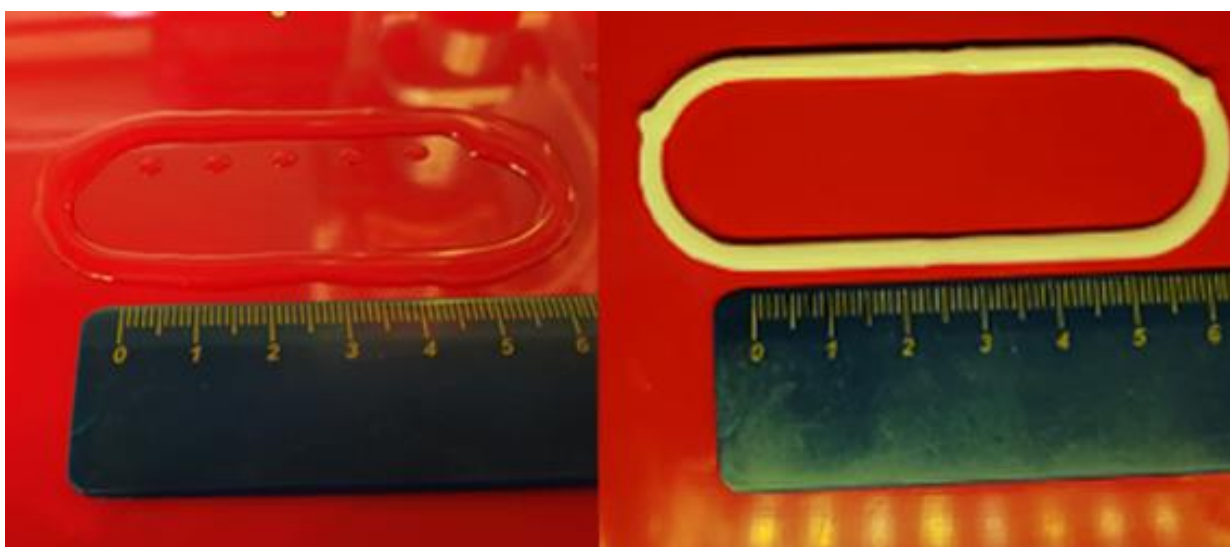
Fonte: Produzida pelo autor.

Para realizar a impressão foi utilizada a bioimpressora *Genesis™* da fabricante 3DBS, uma Seringa Bico *Luer Lock* de 10 ml, uma agulha plástica (tipo de ponteira) cônica de 25G. Os parâmetros usados foram:

- Altura da Camada: 0,05 mm
- Espessura do Bico: 0,1 mm
- Temperatura do Bico: 25°C
- Temperatura da Mesa: 25°C
- Taxa de Alimentação: 10 mm/s
- Retração Habilitada: *True*
- Velocidade de Retração: 25 mm/s
- Distância de Retração: 1 mm
- Distância de Retração (Extrusora): 1 mm
- Espessura do Filamento: 3,0 mm - Apesar de estarmos usando hidrogel, e não filamento na impressão, quanto maior esse parâmetro (espessura do filamento), menor é o deslocamento da extrusora. Conseqüentemente, menor é a quantidade de material dispensado na mesa. É importante ajustar esse parâmetro, pois dependendo da espessura da seringa a quantidade de material dispensado para o mesmo deslocamento da extrusora poderá variar. O valor do parâmetro espessura do filamento para este teste foi definida por tentativa e erro.

A Figura 39 ilustra, à direita, a impressão com hidrogel e, à esquerda a impressão com creme hidratante, ambas com uma régua posicionada abaixo da estrutura impressa. Comparando a geometria do *design* feito digitalmente com os objetos impressos, pode-se notar que a impressão com hidrogel manteve o comprimento das linhas, porém devido à baixa viscosidade da amostra usada no teste, a estrutura colapsou e o material se espalhou para os lados. Também pode-se notar algumas gotas do material no limite interno da linha superior. As gotas são consequência da baixa viscosidade do material e ao baixo rigor no controle da temperatura do material, pois ao deslocar a cabeça de impressão para imprimir a próxima camada o material gotejou ao longo do trajeto. Já na impressão feita com creme hidratante, pode-se notar que as medidas foram mantidas e a linha manteve sua estrutura sem espalhamento lateral. Apesar do espalhamento do material na impressão do hidrogel devido à baixa viscosidade da amostra usada no teste, o *BioSlicer* controlou a impressora mantendo a precisão na deposição de material camada a camada, de acordo com o *design* original.

Figura 39: Testes com Bioimpressora: à esquerda, impressão com hidrogel (alginato 2% e 7% gelatina); à direita impressão com creme hidratante.



Fonte: Produzida pelo autor.

5.3.3 Limitações atuais de uso

A versão do BioSlicer foi testada e homologada somente na impressora 3D *Creality Ender 3 Pro* e na bioimpressora *Genesis™*. Portanto, neste momento, o *software* em outras impressoras pode não funcionar adequadamente. O gCode gerado pelo BioSlicer foi desenvolvido para impressoras com *firmware* da *Marlin*, assim o uso em impressoras com outros tipos de *firmware* pode apresentar resultado indesejado. Além disso o foco do trabalho foi a impressão por extrusão, portanto o uso do BioSlicer em impressoras que utilizam outro processo pode apresentar resultado indesejado ou não funcionar.

5.4 Ambiente de desenvolvimento do *software* proposto

Para desenvolver este trabalho foi necessária a preparação do ambiente de desenvolvimento. Neste ambiente algumas ferramentas de *software* foram instaladas e configuradas. Nesta seção, são descritas as ferramentas utilizadas, bem como sua configuração.

5.4.1 Git

O *plugin* foi integrado ao *software* livre *InVesalius*, como descrito na seção 1.6.4. Para fazer a integração é necessário fazer o *download* do código fonte do *InVesalius* para que se possa fazer as integrações necessárias bem como entender sua estrutura. O código fonte do *InVesalius* está disponível para *download* na plataforma Git.

O Git é um sistema SCM (*Source Code Management*) e é muito usado por desenvolvedores para o controle de versão distribuída de código fonte. O Git é gratuito e de código aberto projetado para lidar com projetos de pequeno, médio e grande porte com velocidade e eficiência (Git-Scm, 2023). Para esse projeto foi usada a versão *Standalone Installer 2.39.0.2* para *Windows 64 Bits* obtida do endereço <https://git-scm.com/download/win> (Acessado em 16 de janeiro de 2023).

Para obter o código fonte do *InVesalius*, após a instalação do pacote, utiliza-se o comando `git clone` <https://github.com/invesalius/invesalius3.git>. Esse comando fará um *download* do código fonte do *InVesalius* para o computador local. Após obter o código fonte as instruções para a instalação do ambiente necessário para compilar o *InVesalius* na máquina local podem ser obtidas no arquivo README.md na pasta *invesalius3* criada após o clone do diretório através da plataforma git.

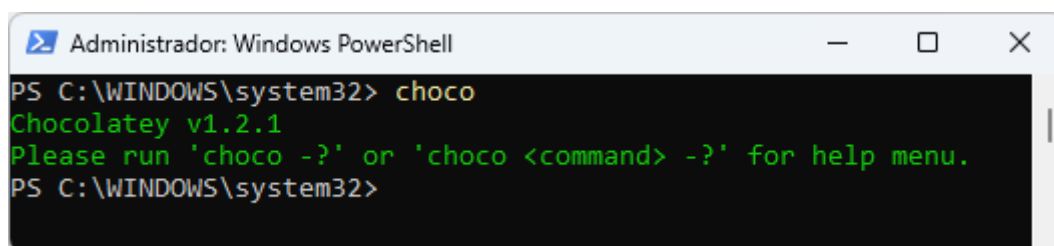
5.4.2 Chocolatey

Alguns algoritmos do *InVesalius* são escritos em *Cython* para melhorar o desempenho. Ao compilar o código fonte também é necessário compilar os algoritmos escritos na linguagem *Cython*. Para isso os desenvolvedores do *InVesalius* usaram a distribuição do *Visual Studio 2019 community edition*. Os desenvolvedores do *InVesalius* recomendam a instalação dos pacotes necessários do *Visual Studio* do repositório *Chocolatey*.

O *Chocolatey* é um gerenciador de pacotes para *Windows*, devido à diversidade de formatos de instalador e às várias abordagens para a implantação do *software* no *Windows*. A implantação de *software* sem gerenciamento de pacotes no *Windows* pode ser complicada e demorada, além de não alcançar o resultado esperado pelo desenvolvedor (Chocolatey, 2023).

A instalação do *Chocolatey* foi feita através do *powershell* do *Windows*. Antes de executar o comando de instalação é necessário liberar a execução de *scripts* externos utilizando o comando `Set-ExecutionPolicy AllSigned` no *prompt* de comando do *powershell*, sendo executado como administrador do sistema. Em seguida, o comando `Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))` foi executado no *prompt* de comando do *powershell*, também como administrador, para instalar o *Chocolatey* no sistema. A versão do *Chocolatey* instalada foi a v1.2.1 conforme demonstrado na Figura 40.

Figura 40: Versão do Chocolate.



```
Administrador: Windows PowerShell
PS C:\WINDOWS\system32> choco
Chocolatey v1.2.1
Please run 'choco -?' or 'choco <command> -?' for help menu.
PS C:\WINDOWS\system32>
```

Fonte: Produzida pelo autor.

As instruções completas para a instalação do *Chocolatey* podem ser visualizadas no website <https://chocolatey.org/install>.

5.4.3 Visual Studio 2019 *community edition*

Conforme citado na seção anterior, a instalação do *Visual Studio 2019 community edition* é necessária para compilar os algoritmos escritos na linguagem *Cython*. O *Visual Studio 2019 community edition* é um IDE (*Integrated Development Environment* - ambiente de desenvolvimento integrado) gratuito da *Microsoft*, multilinguagem e modular (*Microsoft*, 2023). A instalação do *Visual Studio* foi feita usando dois pacotes do *chocolatey*: *visualstudio2019buildtools* e *visualstudio2019-workload-vctools* conforme recomendação dos desenvolvedores do *InVesalius*. Para instalar os pacotes os comandos *choco install -y visualstudio2019buildtools* e *choco install -y visualstudio2019-workload-vctools* devem ser executados, nesta ordem, no *prompt* do *powershell*.

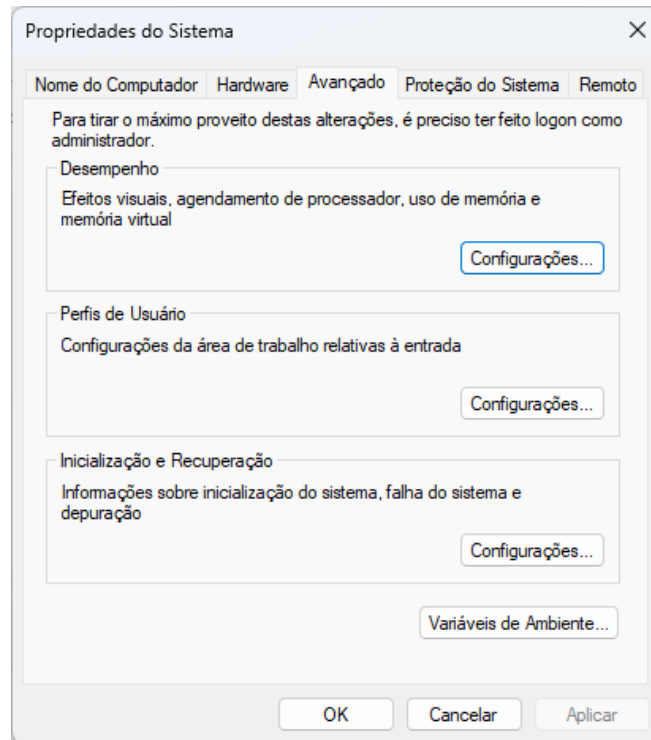
5.4.4 Python

A instalação do compilador da linguagem *Python* é necessária para compilar o *InVesalius* bem com o *plugin* proposto neste trabalho. A versão utilizada nesse trabalho foi a versão 3.8.10 recomendada pelos desenvolvedores do *InVesalius*. O pacote de instalação dessa versão pode ser obtido no site <https://www.python.org/downloads/windows/>.

Após a execução do pacote de instalação é necessário configurar as variáveis de ambientes no *Windows*. Para configurar as variáveis de ambiente o

usuário precisa abrir a janela de sistema no *Windows* e clicar em configurações avançadas do sistema para abrir a janela ilustrada na Figura 41.

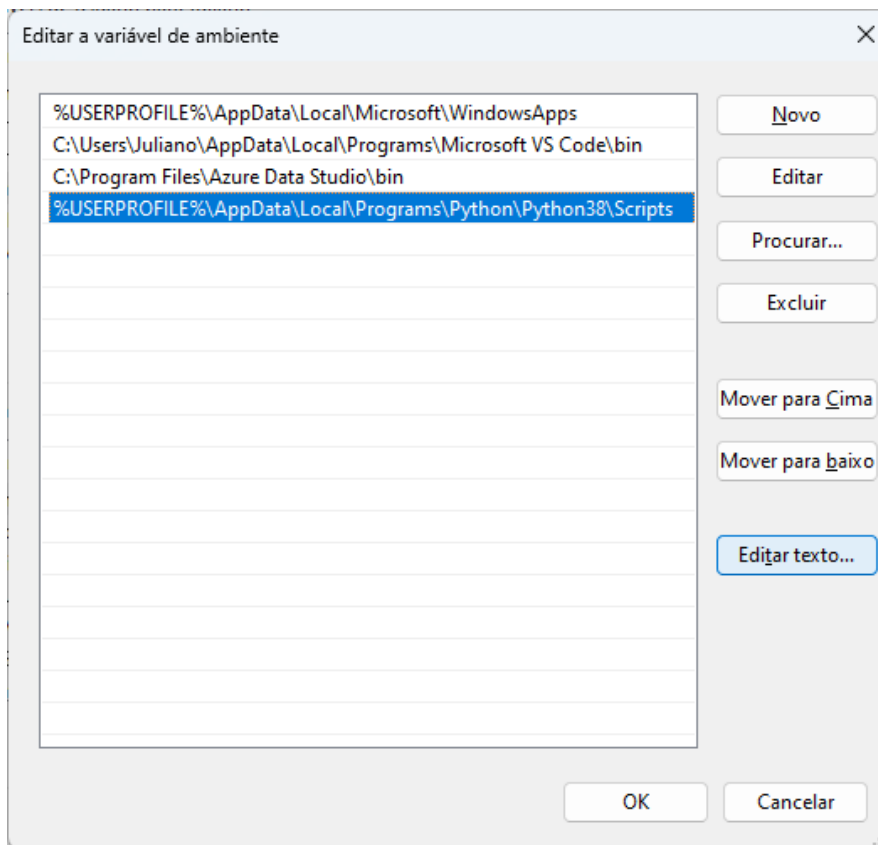
Figura 41: Janela Propriedades do Sistema do Windows 11.



Fonte: Produzida pelo autor.

Em seguida, clicar no botão variáveis de Ambiente e acessar a guia variáveis de usuário. Selecionar a variável *Path*, clicar em editar e adicionar a seguinte linha
`%USERPROFILE%\AppData\Local\Programs\Python\Python38\Scripts`
conforme Figura 42.

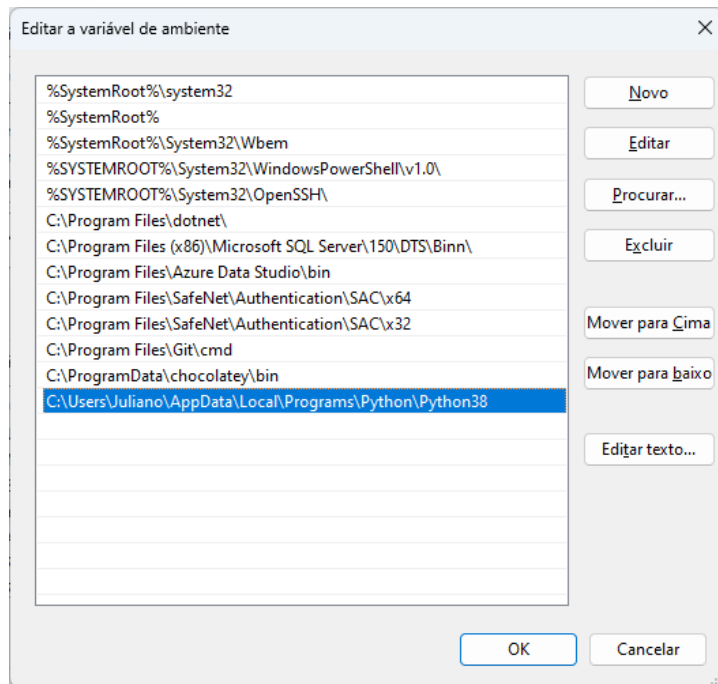
Figura 42: Janela Editar variável de ambiente do usuário.



Fonte: Produzida pelo autor.

Nas variáveis de sistema, selecionar a variável *Path* e adicionar a seguinte linha `C:\Users\Juliano\AppData\Local\Programs\Python\Python38` conforme Figura 43. Substituir o trecho `C:\Users\Juliano\` pelo local onde foi instalado no computador que o *python* está sendo configurado.

Figura 43: Janela Editar variável de ambiente do sistema.



Fonte: Produzida pelo autor.

Após a instalação do *Python* é necessária a instalação das dependências que foram utilizadas no desenvolvimento do *InVesalius*. A versão do *InVesalius* aplicada neste trabalho utiliza as seguintes dependências.

- Cython versão 0.29.28
- Pillow versão 9.3.0
- Pypubsub versão 4.0.3
- configparser versão 5.0.1
- h5py versão 2.10.0
- imageio versão 2.9.0
- nibabel versão 3.2.1
- numpy versão 1.22.1
- plaidml-keras versão 0.7.0
- psutil versão 5.8.0
- pyserial versão 3.5
- python-gdcm versão 3.0.12
- scikit-image versão 0.19.1
- scipy versão 1.7.3
- vtk versão 9.1.0

- wxPython versão 4.1.1
- Theano versão 1.0.5
- torch versão 1.13.1
- pyacvd versão 0.2.7

A instalação das dependências é feita através do pip. O pip é um sistema de gerenciamento de pacotes padrão usado para instalar e gerenciar pacotes de *software* escritos em *Python*. Antes de instalar as dependências é necessário atualizar o pip através do seguinte comando `python -m pip install --upgrade pip` executado no *prompt* de comandos.

No diretório raiz do código fonte do *InVesalius* está disponível o arquivo chamado *requirements.txt* com a lista dos pacotes e sua versão. Para instalar as dependências utilizar o comando no *prompt* de comandos do dos `pip install -r requirements.txt` a partir do diretório raiz do código fonte do *InVesalius*.

No ambiente usado para desenvolvimento deste trabalho também foram instalados os pacotes para ativar os recursos de neuronavegação executando o comando `pip install pyclaron polhemusFT polhemus pypolaris pypolarisP4` e o pacote opcional para computação da tractografia em tempo real durante a neuronavegação executando o comando `pip install https://github.com/dmritrekker/trekker/raw/master/binaries/Trekker-0.9-cp38-cp38-win_amd64.whl` no *prompt* de comandos.

5.4.5 Compilando e executando o *InVesalius*

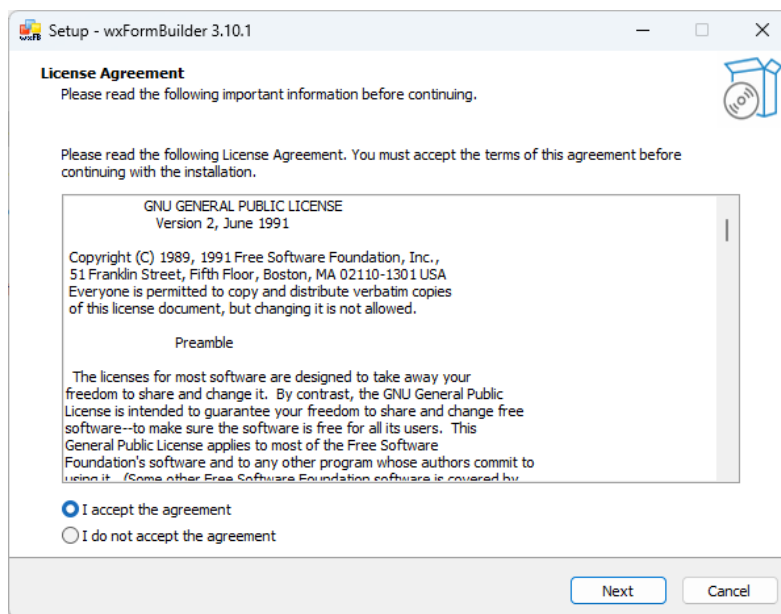
Após a instalações dos pacotes, é possível compilar o *InVesalius*. Para compilar o *InVesalius*, o comando `python setup.py build_ext --inplace` foi executado no *prompt* de comando na pasta raiz do código fonte do *InVesalius*.

Ao término da compilação do código fonte o comando `python app.py` é executado dentro do diretório do código fonte do *InVesalius* para abrir o *software*. As orientações da instalação detalhada do *InVesalius* no *Windows* podem ser obtidas pelo *link* <https://github.com/invesalius/invesalius3/wiki/Running-InVesalius-3-in-Windows>.

5.4.6 wxFormBuilder

Para desenvolver a interface do *plugin* proposto no trabalho foi utilizado o *software* *wxFormBuilder* na versão 3.10.1 para *Windows 64 bits*, obtida do endereço eletrônico <https://github.com/wxFormBuilder/wxFormBuilder/releases>. O *wxFormBuilder* é um construtor de GUI (*Graphical User Interface* - interface gráfica do usuário) para a estrutura *wxWidgets*. Ao executar o pacote de instalação a tela do contrato de licença é exibida, conforme Figura 44.

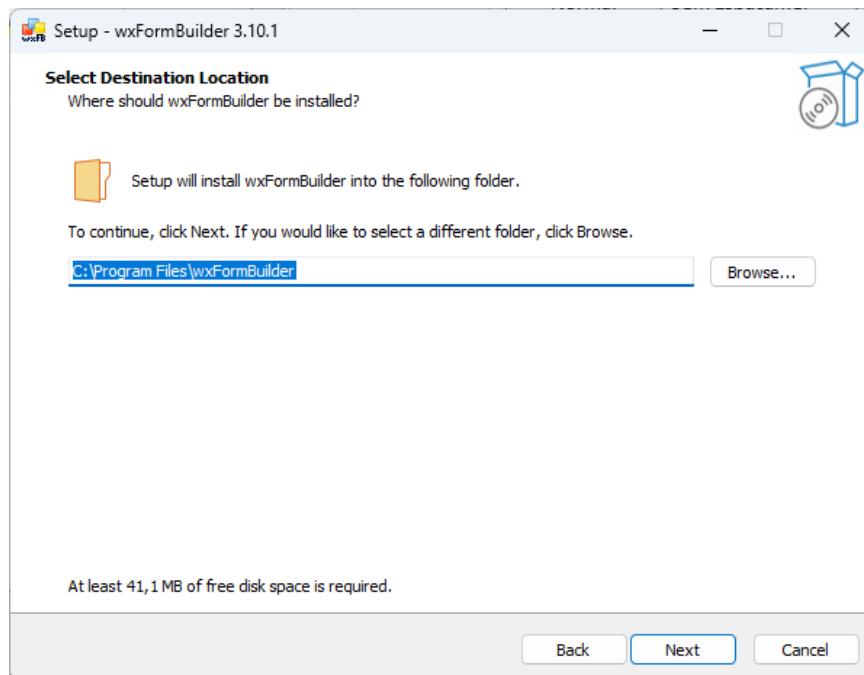
Figura 44: Instalação do wxFormBuilder Licenciamento.



Fonte: Produzida pelo autor.

Após o aceite do contrato de licenciamento deve-se definir a pasta onde o *software* será instalado, conforme ilustrado na Figura 45.

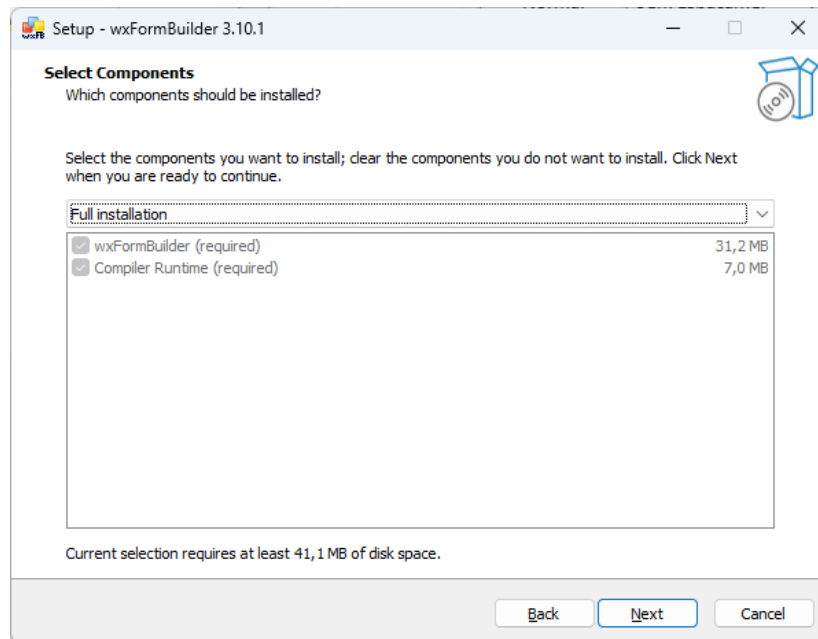
Figura 45: Instalação do wxFormBuilder pasta de destino.



Fonte: Produzida pelo autor.

O próximo passo para a instalação do *wxFormBuilder* é selecionar os componentes a serem instalados. Para o desenvolvimento desse trabalho foi selecionada a opção *Full Installation* (instalação completa) para instalar todos os componentes disponíveis, conforme demonstrado na Figura 46.

Figura 46: Instalação do wxFormBuilder - seleção dos componentes.



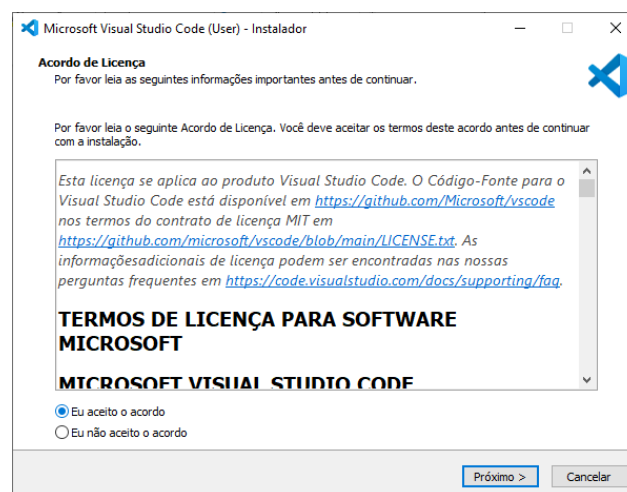
Fonte: Produzida pelo autor.

5.4.7 Visual Studio Code

Para desenvolver os códigos do *plugin* proposto nesse trabalho foi utilizado o *framework Visual Studio Code* na versão 1.74.3 da *Microsoft* para *Windows 64 bits* disponível para *download* pelo endereço eletrônico <https://code.visualstudio.com/download>.

Após o *download* e execução do arquivo de instalação é necessário aceitar o contrato de licença conforme Figura 47.

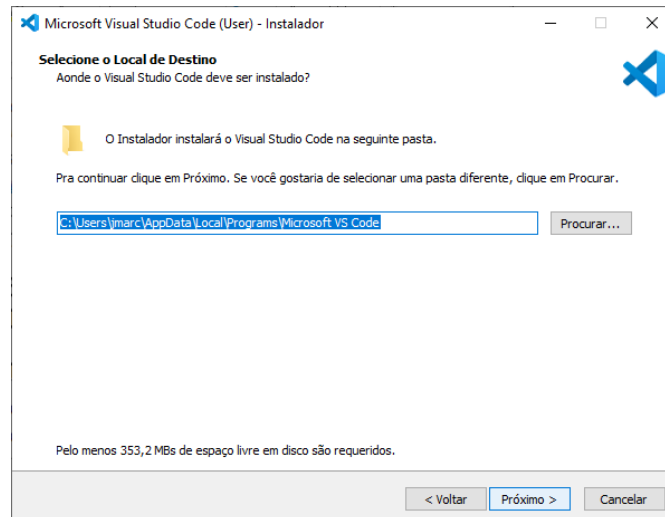
Figura 47: Instalação do Visual Studio Code - Licenciamento.



Fonte: Produzida pelo autor.

O próximo passo é escolher a pasta de destino onde o *Visual Studio Code* será instalado, conforme Figura 48.

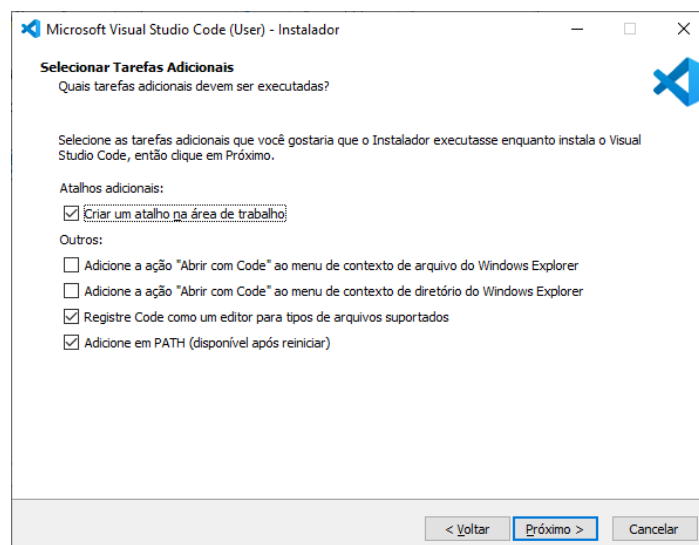
Figura 48: Instalação do Visual Studio Code – Local de Destino.



Fonte: Produzida pelo autor.

Após escolher a pasta de destino é necessário escolher a pasta do Menu Iniciar do *Windows* onde serão adicionados os atalhos do programa. Por fim, é exibida a tela de seleção das tarefas adicionais da instalação do *Visual Studio Code* a serem executadas. A Figura 49 mostra as opções que foram escolhidas na instalação utilizada no desenvolvimento desse trabalho.

Figura 49: Instalação do Visual Studio Code – Tarefas a serem executadas.



Fonte: Produzida pelo autor.

5.4.8 Pacotes adicionais do *Python*

Na linguagem *Python* módulos são programas que tipicamente contêm funções, variáveis, classes e objetos que provêm alguma funcionalidade comum e que podem ser usadas em outros programas. Um pacote (biblioteca) é basicamente outro tipo de módulo que contém um conjunto de módulos. Para o desenvolvimento deste trabalho, além dos pacotes que foram instalados como dependência do *InVesalius*, também foi instalado o pacote *numpy-stl*. O *numpy-stl* é uma biblioteca para trabalhar com arquivos STL (e objetos 3D em geral). A instalação dessa biblioteca foi feita através do comando *pip install numpy-stl*.

6 Discussão e Trabalhos Futuros

A tecnologia de Manufatura Aditiva (impressão 3D) começou a ser usada na área da saúde na década de 80. A utilização de biomateriais na impressão 3D ajudou a consolidar a biofabricação e a bioimpressão 3D e sua evolução é complexa e depende da superação de desafios tecnológicos, biológicos, químicos, físicos, de materiais, éticos e de regulamentação (Freeman *et al.*, 2022).

A qualidade da bioimpressão depende muito dos parâmetros dos materiais com os quais a impressão foi feita. Mudanças em parâmetros como a temperatura, tempo, velocidade de impressão, diâmetro do bico e pressão de distribuição, podem afetar o resultado da bioimpressão (Bahraminasab, 2020).

Um estudo feito para otimizar o processo de bioimpressão de biotintas híbridas, concluiu que a pressão, a velocidade e diâmetro do bico, bem como seu deslocamento e padrão angular, são parâmetros que desempenham funções que afetam a qualidade de impressão (James e Mulgaonkar, 2022). Além disso, em bioimpressoras por extrusão com cabeçotes pneumáticos, a pressão exercida pode causar uma alta tensão de cisalhamento e comprometer a viabilidade celular, dependendo da viscosidade e composição da biotinta. Portanto, otimizar a pressão exercida, bem como outros parâmetros envolvidos na bioimpressão, como temperatura e diâmetro do bico podem garantir a proliferação celular, sem a necessidade e tentativa e erro (Webb e Doyle, 2017).

Pesquisadores estudaram a tensão máxima de cisalhamento da parede, através de simulações de estado estacionário feitas para três geometrias de bico, com diâmetros de bico variados (0,1mm – 0,5mm), em diferentes pressões de entrada (0,025 MPa – 0,25 MPa), como condições de entrada. Concluíram que aumentar o diâmetro do bocal aumenta a taxa de fluxo de massa e diminui a tensão de cisalhamento da parede, mas pode afetar negativamente a resolução da impressão. Também constataram que a pressão mais alta precisa ser aplicada no bocal com menor diâmetro para atingir a mesma vazão dos outros dois bocais, resultando em maior tensão de cisalhamento (Chand, Muhire e Vijayavenkataraman, 2022).

Cada biotinta tem propriedades únicas de material e características de extrusão que podem limitar a precisão da bioimpressão. Devido às particularidades de cada biotinta ao imprimir construções com formatos idênticos, porém com biotintas diferentes, dificilmente teremos construções idênticas (Sedigh, *et al.*,2022). Portanto, os parâmetros de bioimpressão devem ser ajustados de acordo com o biomaterial usado na impressão e a abordagem utilizada para imprimir, seja ela de extrusão, jato de tinta ou a laser.

Por exemplo, Coşkun, Sema *et al.* (2022) desenvolveram uma biotinta otimizada à base de quitosana. Seus estudos mostraram que a formulação pode ser impressa a uma pressão de 50 a 70 kPa e velocidade de impressão de 4 a 11 mm/s, com alta viabilidade celular. Nesse estudo, os pesquisadores usaram a abordagem de impressão por extrusão.

Outro estudo usou sistemas *Fuzzy* para identificar conjuntos de parâmetros de impressão otimizados que melhoram a precisão de impressão de biotintas existentes ou recém-formuladas. Os autores observaram que a otimização usando o sistema *Fuzzy* melhorou a precisão em uma biotinta de colágeno em 15% e na *Pluronic F127* em 68%, em comparação com os parâmetros de impressão recomendados pelo fabricante (Sedigh, *et al.* 2022).

Outros autores, como Bonatti, *et al.*, (2022) propuseram a utilização de *Deep Learning* como ferramenta para otimizar automaticamente os parâmetros de impressão e monitorar o processo de impressão online. Em um ambiente controlado mostraram que é possível utilizar tecnologia de *machine learning* na otimização de parâmetros de bioimpressão.

Na tentativa de obter máxima precisão geométrica na impressão de biotintas em bioimpressoras baseadas em extrusão, os pesquisadores Webb e Doyle (2017) desenvolveram um método simples de avaliar os resultados da bioimpressão de uma variedade de parâmetros de impressão de maneira padronizada. Os pesquisadores criaram o índice de otimização de parâmetros (POI - *parameter optimization index*), que tem por objetivo minimizar a tensão de cisalhamento que atua na biotinta e, conseqüentemente, nas células encapsuladas com a máxima precisão geométrica. O POI sugere os parâmetros ideais de calibre do bico, pressão de impressão e velocidade de impressão.

Segundo Guimarães (2024), embora muito trabalho venha sendo desenvolvido visando melhorar a qualidade da impressão e a função de objetos

bioimpressos, ainda há muito trabalho a fazer. Devido às especificidades dos biomateriais, tais como citocompatibilidade e extrudabilidade, bem como a diversidade de tecnologias de impressão, ao iniciar a bioimpressão com um material específico, muitas vezes é necessária uma investigação completa para entender a influência de cada parâmetro de impressão (Webb e Doyle 2017).

A autora Vaz (2021) em seu trabalho estabeleceu um processo de caracterização da printabilidade de uma biotinta por meio de uma abordagem de design com o desenvolvimento de modelos CAD e pela aplicação de modelos matemáticos e estatísticos para otimização do processo de escolha de parâmetros.

Araújo Neto (2023) usou inteligência artificial para aprendizagem e predição das imagens histológicas dos filamentos de hidrogéis, diferenciando aqueles estirados dos não estirados pela morfologia.

Nesse contexto, e alinhando aos objetivos do trabalho, que além de integrar as etapas do processo de bioimpressão em uma única interface facilitando o processo de uso do *software* pelo usuário, também tem como objetivo integrar ferramentas que ajudem o usuário na otimização dos parâmetros de bioimpressão de acordo com as características dos biomateriais, bem como, novas funcionalidades que possibilitem ao usuário maior controle, precisão e simplicidade de configuração de parâmetros na impressão 3D utilizando biomateriais.

6.1 Sugestões de trabalhos Futuros

Assim como trabalhos a serem desenvolvidos a partir deste, podem-se sugerir:

- I. Criar um algoritmo para integrar o índice de otimização de parâmetros (POI), desenvolvido por Webb e Doyle (2017), ao *plugin* e ajustar os parâmetros de fatiamento de acordo com as características do biomaterial usado;
- II. Adicionar funções para impressão em placa de poços, permitindo que o usuário selecione os poços aos quais ele deseja imprimir sem ter que editar o *gCode*;

- III. Alterar o algoritmo de fatiamento para permitir parâmetros diferentes nas camadas externas e internas do modelo a ser impresso para melhorar a resolução da impressão;
- IV. Criar um visualizador de *gCode* para que se possa inspecionar o *gCode* camada a camada antes de imprimir, bem como, visualizar o percurso da impressão;
- V. Permitir importar múltiplos modelos STL e selecionar parâmetros de impressão diferentes para os modelos importados, permitindo que se possa imprimir, no mesmo projeto, *scaffolds* e biomateriais ou combinar a utilização de materiais diferentes em uma mesma impressão;
- VI. Desenvolver ferramentas que possibilitem a adição e pré-configuração de novas impressoras sem a necessidade de exportar as informações em arquivo .ini;
- VII. Gerar um banco de dados mais complexo e editável para novas inserções a partir da experiência dos próprios usuários.

7 Conclusão

A tecnologia de biofabricação está evoluindo para um complexo sistema composto por muitos processos, incluindo desenho assistido por computador (BioCAD), engenharia assistida por computador (BioCAE), fabricação assistida por computador (BioCAM) e processos biológicos, que dependem da combinação de diferentes componentes inter-relacionados, como moléculas, genes, redes reguladoras, células, organoides e tecidos, integrados com abordagens como projeto, modelagem, simulação e otimização, entre outros.

Nesse contexto os *softwares* são fundamentais para viabilizar a biofabricação, estando presentes em todas as etapas do processo, desde a concepção do modelo a ser impresso e no auxílio da preparação dos biomateriais envolvidos, passando pelo controle dos dispositivos, tais como bioimpressoras e/ou biorreatores e *softwares* de análises e gerenciamento das informações geradas no processo. As perspectivas de bioimpressão não podem ser fomentadas sem o desenvolvimento em paralelo de *software* de computador podendo afirmar que as promessas da bioimpressão andam de mãos dadas com as promessas do *software*. Como visto na literatura, outros autores previram em seus artigos que um único *software* moderno permitirá projetar todo o processo de biofabricação de órgãos e controlar o equipamento de biofabricação robótica correspondente. Nesta conjuntura, a integração dos processos de biofabricação juntamente com complementação de novos processos a *softwares* renomados tem suma importância para a evolução da tecnologia.

O *plugin BioSlicer* desenvolvido nesse trabalho contribui para a evolução da área estendendo as funcionalidades do renomado *software* de análise de imagens médicas desenvolvido pelo CTI Renato Archer denominado *InVesalius*.

Os testes de impressão mostram que as estruturas impressas, tanto utilizando material sólido quanto com material não sólido, mantiveram as características dos modelos originais, mostrando a eficácia do *plugin*. A versatilidade na configuração dos parâmetros e a possibilidade de exportação e importação dos parâmetros contribui para um maior dinamismo na usabilidade do *software*, permitindo que usuários e pesquisadores mantenham um banco de dados com as informações de cada modelo impresso. O *software* também

permite a exportação do *gCode* para a utilização de impressoras que não possuem porta USB, bem como, o envio do *gCode* permitindo a impressão direta do computador via porta USB.

Apesar de cumprir com os objetivos do trabalho ora proposto e estar alinhado com as demandas da área de biofabricação, melhorias devem ser implementadas para se alcançar o objetivo de uma ferramenta altamente personalizada com as necessidades da bioimpressão de tecidos e órgão.

Para contribuir com comunidade acadêmica, o *software* foi registrado no INPI e seu código disponibilizada para que outros pesquisadores possam contribuir com melhorias e a adição de novas funcionalidades. O código fonte ficará disponível, no período de dois anos, no endereço eletrônico https://uniara-my.sharepoint.com/:f:/g/personal/jmarcello_uniara_com_br/EhvJxmHxILVDrHMZIA1ylyEBGpKwoh03wyaQSi7xu6Aczw?e=He9Twr.

8 Referências Bibliográficas

3DLAB. **O que é um arquivo STL?**. Disponível em: <https://3dlab.com.br/afinal-o-que-e-um-arquivo-stl/> Acesso em: 23 de Out. de 2020.

3DSYSTEMS. **O que é um arquivo STL?**. Disponível em: <https://br.3dsystems.com/quickparts/learning-center/what-is-stl-file> Acesso em: 23 de Out. de 2020.

AMORIM P. H. J.; Moraes, T. F.; Azevedo, F. S.; Silva, J. V. L. **InVesalius: Software Livre de Imagens Médicas**. XXXI Congresso da Sociedade Brasileira de Computação, 2011. p. 1735-1740.

ARAÚJO NETO, L. A. **Filamentos de hidrogéis à base de polímeros para a incorporação de células : avaliação reológica, biomecânica e uso de aprendizado de máquina**. 2023. Tese (Doutorado) - Universidade Federal do Paraná, Paraná, 2023.

ASTM F2915-12, **Standard Specification for Additive Manufacturing File Format (AMF) Version 1.1** (Withdrawn 2013), ASTM International, West Conshohocken, PA, 2012, www.astm.org

BAHRAMINASAB, Marjan. **Challenges on optimization of 3D-printed bone scaffolds**. BioMedical Engineering OnLine, v. 19, n. 1, p. 1-33, 2020.

BICUDO, E., Faulkner, A., Li, P. **Digital readiness in 3D bioprinting: software, governance and hospitals' proto-clinical interfaces**. Future Medicine, 2021 doi: <https://doi.org/10.2217/rme-2020-0132>

BCN3D. **Folha de referências do gCode para impressão 3D: demandas básicas**. Disponível em: < <https://www.bcn3d.com/3d-printer-g-code-demands-basic-guide/>> Acessado em: 24 de outubro de 2020.

BONATTI, Amedeo Franco et al. **A Deep Learning Quality Control Loop of the Extrusion-Based Bioprinting Process**. International Journal of Bioprinting, v. 8, n. 4, 2022.

BROCK, Thomas Dale *et al.* **Brock biology of microorganisms**. Upper Saddle River (NJ): Prentice-Hall, 2003., 2003.

BOURKE, Paul. **STL format**. Outubro de 1999. Disponível em: <http://paulbourke.net/dataformats/stl/>. Acesso em: 15 de Fev. de 2023.

BUSTAMANTE, R. M. **Fatiador AMF com subdivisão de triângulos curvos e ajuste de curvas**. Tese (Mestrado em computação aplicada) - Centro de Ciências e Tecnologia, Universidade do Estado de Santa Catarina. Joinville, p. 45-46. 2019.

CAPES, 2019. **Documento de Área**, Área 48, Biotecnologia, Ministério da Educação.

CHAND, Rashik; MUHIRE, Beni Shimwa; VIJAYAVENKATARAMAN, Sanjairaj. **Computational Fluid Dynamics Assessment of the Effect of Bioprinting Parameters in Extrusion Bioprinting**. International Journal of Bioprinting, v. 8, n. 2, 2022.

CHOCOLATEY. **The Package Manager for Windows**. Disponível em: <https://chocolatey.org/>. Acesso em: 16 de Jan. de 2023.

CAMPOS, F. L. M. **Desenvolvimento de uma interface para comando e monitorização de Impressoras 3D no âmbito da indústria 4.0**. Dissertação (Mestrado em Engenharia) - Faculdade de Engenharia, Universidade do Porto. Porto, Portugal. p. 17. 2019.

COŞKUN, Sema et al. **Formulation of chitosan and chitosan-nanoHAp bioinks and investigation of printability with optimized bioprinting parameters**. International Journal of Biological Macromolecules, v. 222, p. 1453-1464, 2022.

CTI. **InVesalius 3**. Disponível em: <https://www.cti.gov.br/pt-br/invesalius>. Acesso em: 24 de Out. de 2020.

BELO, F. L. **Manufatura aditiva de biocompósitos a partir de ácido poliláctico reforçado por hidroxiapatita e nanotubos de carbono para regeneração de tecido ósseo**. Repositório da ufpa.br, 2024.

Dávila, JL, Freitas, MS, Neto, PI et al. **Software para gerar caminhos de impressão 3D contínuos para a fabricação de andaimes de engenharia de tecidos**. Int J Adv Manuf Technol 84, 1671–1677 (2016). <https://doi.org/10.1007/s00170-015-7866-8>

DAVIS. **vtkRenderer Class Reference**. Disponível em: <http://davis.lbl.gov/Manuals/VTK-4.5/classvtkRenderer.html>. Acesso em: 29 de Jan. de 2023.

DERNOWSEK, J. A., REZENDE, R. A., SILVA, J. V. L. **The role of information technology in the future of 3D biofabrication**. J. 3D Print. Med. (2017-a) 1(1), 63–74

DERNOWSEK J.A., REZENDE, R.A., SILVA J.V.L. (2017-b) **BioCAE: A New Strategy of Complex Biological Systems for Biofabrication of Tissues and Organs**. J Tissue Sci Eng 8: 200. doi: 10.4172/2157-7552.1000200

DERNOWSEK, J. A.; REZENDE, R. A.; SILVA, J. V. L. **Biofabricação e suas estratégias**. VIRUS, São Carlos, n. 11, 2015. Disponível em: <http://www.nomads.usp.br/virus/virus11/?sec=5&item=60&lang=pt>. Acesso em: 24 de Out. 2022.

DERNOWSEK, Janaina. **Desafios da Bioimpressão**. biofabricacao.com, 2018. Disponível em: <https://www.biofabricacao.com/>.>. Acesso em: 09/08/2022.

DICOM. **Part 1: Introduction and Overview.** Disponível em: <http://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>. Acesso em: 23 de Out. de 2020.

DICOM. **Part 10: Media Storage and File Format for Media Interchange.** Disponível em: <http://dicom.nema.org/medical/dicom/current/output/pdf/part10.pdf>. Acesso em: 23 de Out. de 2020.

DICOM. **Part 3: Information Object Definitions.** Disponível em: <http://dicom.nema.org/medical/dicom/current/output/pdf/part03.pdf>. Acesso em: 23 de Out. de 2020.

DICOM. **Part 5: Data Structures and Encoding.** Disponível em: <http://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf>. Acesso em: 23 de Out. de 2020.

DICOMSTANDARD. **Digital Imaging and Communications in Medicine.** Disponível em: <https://www.dicomstandard.org/current/>. Acesso em: 23 de Out. de 2020.

FRANCISCO, L.A.V. **Modelo computacional de descrição de projetos para impressão de biosistemas.** Dissertação (Mestrado em Computação) Universidade Federal de São Carlos. São Carlos, p. 68-69. 2016.

FRANCISCO, Luiz Angelo Valota; BRANDAO, Alexandre Fonseca; DIAS, Diego Roberto Colombo; TREVILIN, Luis Carlos. **Desenvolvimento de projetos para impressão de biosistemas em STL utilizando VTK.** Tendências e Técnicas em Realidade Virtual e Aumentada, [S. l.], p. 5-121, 1 maio 2015. Disponível em: https://www.researchgate.net/figure/Figura-142-Invesalius-Software-Publico-de-Imagens-Medicas-Fonte_fig5_278964932. Acesso em: 22 Jan. 2022.

FREEMAN S, Calabro S, Williams R, Jin S and Ye K (2022) **Bioink Formulation and Machine Learning-Empowered Bioprinting Optimization.** Front. Bioeng. Biotechnol. 10:913579. doi: 10.3389/fbioe.2022.913579

FUSTINONI, D. F. R., FERNANDES, F. C., Leite, F. N. **Informática básica para o ensino técnico profissionalizantes.** Editora IFB, Brasília-DF, 2013.

GARTNER IT GLOSSARY. Disponível em: www.gartner.com/it-glossary/itinformation-technology. Acesso em: 30 de Out. de 2020.

Gartner's 2015 **Hype Cycle for Emerging Technologies.** www.gartner.com/smarterwithgartner/whats-new-ingartners-hype-cycle-for-emerging-technologies-2015. Acesso em: 30 de Out. de 2020.

GIBSON *et al.* **Additive manufacturing technologies: 3D printing, rapid prototyping, and direct digital manufacturing.** second edition. 10.1007/978-1-4939-2113-3, 2015.

GIT-SCM. **Git.** Disponível em: <https://git-scm.com/>. Acesso em: 16 de Jan. de 2023.

GRIMM, Todd. **User's guide to rapid prototyping**. Society of Manufacturing Engineers, 2004.

GROLL *et al.* **Biofabrication: reappraising the definition of an evolving field**. *Biofabrication*. 2016 Jan 8;8(1):013001. doi: 10.1088/1758-5090/8/1/013001. PMID: 26744832.

GROLL *et al.* **A definition of bioinks and their distinction from biomaterial inks**. *Biofabrication* 11 013001, 2018.

GUILLEMOT, F. *et al.* **Bioprinting is coming of age: report from the International Conference on Bioprinting and Biofabrication in Bordeaux (3B'09)**. *Biofabrication* 2, 010201, 2010.

GUIMARÃES, Stephany Castor dos Santos. **Impressão 3D para fins de reparo ósseo: uma revisão da literatura**. 2024.

GUNGOR-OZKERIM, P. S., Inci, I., Zhang, Y. S., Khademhosseini, A., Dokmeci, M. R. **Bioinks for 3D bioprinting: an overview**. *Biomaterials Science*. Issue 5. 2018

ISO/ASTM 52915. **Standard Specification for Additive Manufacturing File Format (AMF) (2013)**.

JAKAB, K., NOROTTE, C., MARGA, F., MURPHY, K., VUNJAK-NOVAKOVIC, G. and FORGACS, G., 2010. **Tissue Engineering by Self-Assembly and Bioprinting of Living Cells**. *Biofabrication*, 2, pp.1-14.

JAMES, Sagil & Mulgaonkar, Samir. (2022). **Study on parameter optimization of 3D bioprinting of hybrid bio-inks**. *The International Journal of Advanced Manufacturing Technology*. 119. 1-12. 10.1007/s00170-021-08561-7.

KENNY A. van Kampen *et al.* **Biofabrication: From Additive Manufacturing to Bioprinting**. *Encyclopedia of Tissue Engineering and Regenerative Medicine*, Academic Press, 2019, Pages 41-55, ISBN 9780128137000.

KIRSCHMAN, C.; JARA-ALMONTE, C., **A parallel slicing algorithm for solid freeform fabrication processes**. *Solid Freeform Fabrication Symposium*, p. 26 - 33, 1992.

LANGER R and VACANTI J P 1993 **Tissue engineering Science** 260 920–6

LANZA, R. , LANGER, R., and VACANTI, J., **Principles of Tissue Engineering**. Burlington, MA, USA: Elsevier, 2007.

LIPSON, Hod. **AMF Tutorial: The Basics (Part 1)**. *3D Printing and Additive Manufacturing v. 1, n. 2*, p. 85–87, jun. 2014. Acesso em: 24 de outubro de 2020.

LIVESU, M.; ELLERO, S.; MARTÍNEZ, J.; LEFEBVRE, S.; ATTENE, M., **From 3D models to 3D prints: an overview of the processing pipeline**. *Computer Graphics Forum*, v. 36, n. 2, p. 537 - 564, 2017.

LONG Ng, W. *et al.* 2018 **Biofabrication** 10 025005

MALONE, E. and LIPSON, H. **Fab@Home: the personal desktop fabricator kit.** Rapid Prototyping Journal. 13/4 (2007) 245–255.

MAO, A. S. and MOONEY D. J. **Regenerative medicine: Current therapies and future directions.** PNAS Special Feature Perspective, vol. 112, n.47, 2015.

MARLIN. **What is Marlin?** Disponível em: <http://marlinfw.org/docs/basics/introduction.html#how-marlin-works>. Acesso em: 04 Fev. 2023.

MIRONOV V., V. KASYANOV, C. DRAKE, R.R. MARKWALD. **Organ printing: promises and challenges.** Regen Med. Jan;3(1) (2008) 93-103.

MIRONOV V, VISCONTI RP, KASYANOV V, FORGACS G, DRAKE CJ, MARKWALD R.R. **Organ printing: tissue spheroids as building blocks.** Biomaterials (2009-a); 30(12):2164–74

MIRONOV, V., J. ZHANG, C. GENTILE, K. BRAKKE, T. TRUSK, K. JAKAB, G. FORGACS, V. KASYANOV, R. P. VISCONTI & R. R. MARKWALD (2009-b): **Designer ‘blueprint’ for vascular trees: morphology evolution of vascular tissue constructs,** Virtual and Physical Prototyping, 4:2, 63-74.

MIRONOV, V., KASYANOV, V., MARKWALD, R. **Organ printing: from bioprinter to organ biofabrication line,** Current Opinion in Biotechnology, 22:1-7, 2011.

MONTERO, F.E, REZENDE, R.A., da SILVA J.V.L. and SABINO M.A. (2019) **Development of a Smart Bioink for Bioprinting Applications.** Front. Mech. Eng. 5:56. doi: 10.3389/fmech.2019.00056.

MORONI, L. *et al.* 2018 **Biofabrication: a guide to technology and terminology** Trends Biotechnol. 36 384–402.

MURPHY,S.V., ATALA, A., 2014. **3D bioprinting of tissues and organs.** Nat. Biotechnol. 32 (8), 773–785.

OZBOLAT IT (2015) **Bioprinting scale-up tissue and organ constructs for transplantation.** Trends Biotechnol 33(7):395–400.

PAKHOMOVA C, POPOV D, MALTSEV E, *et al.*, 2020, **Software for Bioprinting,** Int J Bioprint, 6(3): 279.DOI: 10.18063/ijb.v6i3.279.

PYTHON. **Python.** Disponível em: <https://www.python.org/>. Acesso em: 26 de Out. de 2020.

RAYNA, T. and STRIUKOVA, L., ‘**From Rapid Prototyping to Home Fabrication: How 3D Printing Is Changing Business Model Innovation**’ (2016) Technological Forecasting & Social Change 102, 214–224.

REPETIER (2023). **Repetier-Firmware Documentation**. Disponível em: <https://www.repetier.com/documentation/repetier-firmware/repetierfirmware-introduction/>. Acesso em: 04 de Fev. de 2023.

REPRAP (2023a). **Sprinter**. Disponível em: <https://reprap.org/wiki/Sprinter>. Acesso em: 04 de Fev. de 2023

REPRAP (2023b). **Teacup Firmware**. Disponível em: https://reprap.org/wiki/Teacup_Firmware. Acesso em: 04 de Fev. de 2023.

REPRAP (2023c). **RepRap Firmware**. Disponível em: https://www.reprap.org/wiki/RepRap_Firmware. Acesso em: 04 de Fev. de 2023.

REZENDE RA, PEREIRA FDAS, KEMMOKU BDT *et al.* **Enabling technologies for robotic organ printing**. In: Innovative Developments in Virtual and Physical Prototyping: Proceedings of the 5th International Conference on Advanced Research in Virtual and Rapid Prototyping. Bartolo PJS (Ed.). CRC Press, Leiria, Portugal, 121 (2011).

REZENDE RA, PEREIRA FDAS, KASYANOV V., OVSIANIKOV A., TORGENSEN J, GRUBER P, STAMPFL J, BRAKKE K, NOGUEIRA J, MIRONOV V, SILVA JVL. **Design, physical prototyping and initial characterisation of 'lockyballs'**, Virtual and Physical Prototyping, 7:4, 287-301, DOI: 10.1080/17452759.2012.740877 (2012).

REZENDE RA, KASYANOV V, MIRONOV V, SILVA JVL. **Organ printing as an information technology**. Procedia Eng. 110, 151–158 (2015).

SAMPAIO, S. C. **Modelagem e implementação orientada a objetos de um cliente de rede para banco de dados de imagens médicas digitais utilizando o padrão DICOM 3.0**. Dissertação (Mestrado em Computação) UFSC. Santa Catarina, p. 18-30, 2019.

SEDIGH, Ashkan *et al.* **Enhancing precision in bioprinting utilizing fuzzy systems**. Bioprinting, v. 25, p. e00190, 2022.

SILVA, J.V.L.; REZENDE, R.A. **Additive Manufacturing: Challenges for New Materials Development for Biomedical Applications**. Revista SAM, v. 1, p. 28-41, 2013.

SILVA, Jorge Vicente Lopes da and DUAILIBI, Silvio Eduardo. **A biofabricação de tecidos e órgãos**. ComCiência [online]. 2008, n.102, pp. 0-0. ISSN 1519-7654.

SIMPLIFY3D. **G-Code Tutorial**. Disponível em: <https://www.simplify3d.com/support/articles/3d-printing-gcode-tutorial/>. Acesso em: 24 de Out. de 2020.

SKALAK, R. and FOX, C., **Tissue Engineering**. Liss, New York, 1988.

SQLITE. **What Is SQLite?** Disponível em: <https://www.sqlite.org/index.html> . Acesso em: 26 de Out. de 2020.

TEIXEIRA, Simão Pedro Barbosa. **Molecular imprinting technologies for growth factor sequestration and control of stem cell fate in tendon regeneration.** 2024.

US DHHS Department of Health and Human Services 2006 **A new vision—a future for regenerative medicine,** available at: (<http://medicine.osu.edu/regenerativemedicine/documents/2020vision.pdf>)

US NTSC National Science and Technology Council **2007 Advancing tissue science&engineering: a multi-agency strategic plan** available at: ([https://whitehouse.gov/sites/default/files/microsites/ostp/advancing tissue science-engineering.pdf](https://whitehouse.gov/sites/default/files/microsites/ostp/advancing_tissue_science-engineering.pdf))

VAZ, Gabriela Mendes da Rocha. **Printabilidade de biotinta para processos de bioimpressão 3D por extrusão: abordagem de design e parâmetros de impressão.** 2021. 91 f., il. Dissertação (Mestrado em Biologia Molecular) — Universidade de Brasília, Brasília, 2021.

VEIT, D. R. **Impactos da manufatura aditiva nos sistemas produtivos e suas repercussões nos critérios competitivos.** Tese (Doutorado) - Universidade do Vale do Rio dos Sinos. Programa de Pós-Graduação em Engenharia de Produção e Sistemas, 2018.

VISSER, J., BENJAMIN Peters, THIJS J Burger, Jelle Boomstra, Wouter J A Dhert, Ferry P W Melchels and Jos Malda. **Biofabrication of multi-material anatomically shaped tissue constructs.** *Biofabrication* 5 (2013) 035007 (9pp).

MICROSOFT. **Visual Studio Community.** Disponível em: <<https://visualstudio.microsoft.com/pt-br/vs/community/>>. Acesso em: 16 de Jan de 2023.

MICROSOFT. **Getting Started.** Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 26 de Out. de 2020.

VOLPATO, Neri. **Prototipagem rápida: tecnologias e aplicações.** Editora Blucher, 2007.

VOLPATO, R. **Manufatura Aditiva Tecnologias e aplicações da impressão 3D,** Editora Blucher, São Paulo, 2017.

VTK. **Overview.** Disponível em: <https://vtk.org/about/#overview>. Acesso em: 28 de Jan. de 2023.

WEBB, B., & DOYLE, B. J. (2017). **Parameter optimization for 3D bioprinting of hydrogels.** *Bioprinting*, 8, 8-12. <https://doi.org/10.1016/j.bprint.2017.09.001>

WXPYTHON. **Overview of wxPython.** Disponível em: <https://www.wxpython.org/pages/overview/>. Acesso em: 28 de Jan. de 2023.

APÊNDICE A – DEFINIÇÕES DO PADRÃO DICOM

- **Conformidade:** define princípios para a implementação que reivindica a conformidade com o Padrão.
 - **Requisitos de Conformidade:** especificação dos requisitos gerais.
 - **Declaração de Conformidade:** especificação da informação que deve estar presente.
- **Definições de Objeto de Informação:** as Classes de Objetos de Informação que fornecem uma visão abstrata das entidades do mundo real aplicáveis na comunicação de imagens médicas digitais e suas informações relacionadas (DICOM Part 3, 2020).
- **Especificação das Classes de Serviço:** associam um ou mais objetos de informação através de um ou mais comandos realizados sobre eles. Suas especificações ditam os requisitos tanto para provedores como para usuários dos serviços de comunicação, dos elementos de comandos e como eles são aplicados nos objetos.
- **Estruturas de dados e codificação:** Especifica como aplicações constroem e codificam a informação dos Conjuntos de Dados. Além de definir as regras necessárias para formar o Fluxo de Dados (coleção de Elementos de Dados que compõem o Conjunto) quando uma mensagem é transmitida (DICOM Part 5,2020).
- **Dicionário de Dados:** Define a coleção dos Elementos de Dados disponíveis para representar informações.
- **Troca de mensagens:** Especifica tanto o serviço quanto o protocolo usado por uma aplicação para troca de Mensagens, que é composta de um *Command Stream* seguido por um *Data Stream*.
- **Suporte à comunicação em rede para trocar mensagens:** Especifica os serviços de comunicação e protocolos de camada superior necessários ao suporte, em ambientes de rede, à comunicação eficiente e coordenada entre aplicações DICOM. Onde os serviços são especificados pelo modelo OSI (*Open System Interconnection*) conjuntamente com o protocolo de transporte TCP/IP (*Transmission Control Protocol/Internet*

Protocol), permitindo que aplicativos de pares estabeleçam associações, enviem mensagens e terminem a associação.

- **Armazenamento em mídia e formato de arquivo:** Descreve como deve ser o armazenamento de informações de imagens médicas em mídia removível.
- **Perfil da aplicação do armazenamento em mídia:** Perfis que especificam subconjuntos específicos (Perfil de Aplicação) de aplicativos do Padrão DICOM cuja implementação pode reivindicar a conformidade.
- **Formato de mídia e mídia física para transporte:** Especifica o intercâmbio de informações entre aplicações de sistemas médicos e determina a estrutura de descrição do relacionamento entre o modelo de armazenamento em mídia e uma mídia física específica e formato de mídia.
- **Função de exibição da escala de cinza:** Padroniza as funções para uma exibição consistente de imagens em escala de cinza.
- **Perfis de segurança e sistema de gerenciamento:** Define os perfis de segurança e sistema de gerenciamento para implementações que podem exigir conformidade, utilizando protocolos externos.
- **Recurso de mapeamento de conteúdo:** Especifica modelos (*templates*) de estruturação de documentos como Objetos de Informação DICOM, conjunto de termos codificados para uso nos mesmos objetos, dicionário de termos definidos e mantidos pelo DICOM e traduções para várias línguas desses termos.
- **Informação explicativa:** Define anexos normativos e informativos que contém informações explicativas.
- **Acesso web a objetos persistentes:** Dita os meios pelos quais os Serviços Web podem ser utilizados para recuperar e armazenar objetos persistentes.
- **Hospedagem da aplicação:** Parte do DICOM que especifica uma Interface de Programação de Aplicação (API) para um sistema de computação médico.
- **Relatórios de imagens usando a arquitetura de documentos clínicos HL7:** Especifica modelos (*templates*) para codificação de relatórios de

imagens que usam o padrão HL7 *Clinical Document Architecture Release* (CDA). Nesse escopo, estão os procedimentos clínicos de especialidades que utilizam as imagens para triagem, diagnóstico ou fins terapêuticos. Além disso, contém um guia para implementação do CDA e está harmonizado com a abordagem de modelos padronizados de guias para o CDA desenvolvidos pelo HL7.

APÊNDICE B - INPI – DOCUMENTO DO REGISTRO DO SOFTWARE



REPÚBLICA FEDERATIVA DO BRASIL
MINISTÉRIO DO DESENVOLVIMENTO, INDÚSTRIA, COMÉRCIO E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS

Certificado de Registro de Programa de Computador

Processo Nº: **BR512023003424-0**

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 01/11/2023, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: BioSlicer

Data de publicação: 01/11/2023

Data de criação: 01/11/2023

Titular(es): ASSOCIAÇÃO SÃO BENTO DE ENSINO

Autor(es): JULIANO MARCELLO; RODRIGO ALVARENGA REZENDE; ANDRÉ CAPALDO AMARAL

Linguagem: PYTHON

Campo de aplicação: CO-02; SD-08; SD-09

Tipo de programa: AP-02; AP-03; AT-01; AT-06; AT-07; IT-03; SM-04; TC-01; TC-04

Algoritmo hash: SHA-512

Resumo digital hash:
2e3a171278a37a3db1640ef2e6dc177f4edf31054875c5e6fbd2be8c37e8917e2d99f84476a2980693e3f4d04f5215727bc
0fe0e469af690e965c20958d00be5

Expedido em: 14/11/2023

Aprovado por:
Carlos Alexandre Fernandes Silva
Chefe da DIPTO