



Universidade de Brasília – UnB
Faculdade de Ciências e Tecnologias em Engenharia – FCTE
Engenharia de Software

***In Silico*: Desenvolvimento de um Pâncreas
Artificial para Diabéticos Tipo 1 com Lógica
Difusa e Aprendizado Profundo**

Autor: Kauã Vinícius Ponte Aguiar
Orientador: Prof^(a) Dr(a). Milene Serrano
Coorientador: Prof Dr. Mauricio Serrano

Brasília, DF
2025



Kauã Vinícius Ponte Aguiar

***In Silico*: Desenvolvimento de um Pâncreas Artificial para
Diabéticos Tipo 1 com Lógica Difusa e Aprendizado
Profundo**

Monografia submetida ao curso de graduação
em Engenharia de Software da Universidade
de Brasília, como requisito parcial para ob-
tenção do Título de Bacharel em Engenharia
de Software.

Universidade de Brasília – UnB

Faculdade de Ciências e Tecnologias em Engenharia – FCTE

Orientador: Prof(^a) Dr(a). Milene Serrano

Coorientador: Prof Dr. Mauricio Serrano

Brasília, DF

2025

Kauã Vinícius Ponte Aguiar

In Silico: Desenvolvimento de um Pâncreas Artificial para Diabéticos Tipo 1 com Lógica Difusa e Aprendizado Profundo/ Kauã Vinícius Ponte Aguiar. – Brasília, DF, 2025-

147 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof^(a) Dr(a). Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade de Ciências e Tecnologias em Engenharia – FCTE , 2025.

1. Lógica Difusa. 2. Inteligência Artificial. I. Prof^(a) Dr(a). Milene Serrano.
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. *In Silico*: Desenvolvimento de um Pâncreas Artificial para Diabéticos Tipo 1 com Lógica Difusa e Aprendizado Profundo

CDU 02:141:005.6

Kauã Vinícius Ponte Aguiar

***In Silico*: Desenvolvimento de um Pâncreas Artificial para Diabéticos Tipo 1 com Lógica Difusa e Aprendizado Profundo**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 09 de Dezembro de 2025:

Prof^(a) Dr(a). Milene Serrano
Orientador

Prof Dr. Mauricio Serrano
Coorientador

Prof. Dr. Ronni Geraldo Gomes de Amorim
Examinador 1

Wesley Pereira Araujo
Examinador 2

Brasília, DF
2025

Dedico este trabalho à minha irmã, Kauany, aquela que ilumina meus dias.

Agradecimentos

Antes de tudo, agradeço a Deus e à Santíssima Virgem Maria por terem guiado meus passos até aqui, iluminando meu caminho mesmo nos momentos mais difíceis.

Agradeço primeiramente aos meus pais, Catiane e José, e à minha irmã, Kauany, que sempre estiveram ao meu lado, oferecendo amor, apoio e ensinamentos, que levarei por toda a vida. Obrigado por cada momento. Agradeço também aos meus orientadores, professora Milene e professor Maurício, pela paciência, dedicação e pelos ensinamentos que me fizeram crescer não apenas como estudante, mas também como ser humano.

Também sou grato à Doutora Adriana Moser, a qual sugeriu o tema desta monografia, e me despertou um novo olhar a respeito da Diabetes.

Agradeço aos meus colegas de faculdade que compartilharam essa jornada comigo, em especial à Carla, com quem construí uma amizade verdadeira, e que fez toda a diferença ao tornar a graduação mais leve e divertida. Falando em amizades verdadeiras, sou grato também ao Bruno Henrique pelas maratonas de programação, pelos jogos, e pelas conversas sobre código. Meus agradecimentos ao Eduardo e ao Augusto, que me ajudaram a encontrar ideias para o trabalho quando eu precisava. Também quero agradecer ao Victor, pela ajuda com o material de Engenharia Eletrônica.

Obrigado também à Amanda, que sempre foi como uma irmã mais velha pra mim. Agradeço por me aconselhar e oferecer apoio quando eu mais precisava.

Sou muito grato a Deus por ter colocado pessoas como vocês na minha vida. Obrigado a todos!

"O amor é a única coisa que somos capazes de perceber que transcende as dimensões do tempo e do espaço." (Interestelar, 2014)

Resumo

A Diabetes Mellitus tipo 1 é uma doença de origem autoimune que destrói as células beta pancreáticas produtoras de insulina. Dessa forma, torna-se essencial a reposição exógena do hormônio de maneira vitalícia. Todavia, esse é um processo bastante complexo, que pode levar o paciente a desenvolver ansiedade ou depressão, devido à variedade de estratégias disponíveis para o controle da doença. Nesse cenário, surgem os sistemas de malha fechada, que mimetizam a função do pâncreas humano. Para que esses sistemas funcionem adequadamente, faz-se necessário escolher uma estratégia eficaz para a operação do controlador. Dentre essas estratégias, destacam-se a lógica difusa e a Inteligência Artificial. A lógica difusa permite lidar com a incerteza e a imprecisão dos dados fisiológicos, possibilitando decisões mais flexíveis. Já a Inteligência Artificial, através de uma Rede Neural, oferece a capacidade de aprendizado e adaptação a diferentes perfis de pacientes, o que potencializa a personalização do tratamento. Com isso, por meio do desenvolvimento de provas de conceito (PoCs), este trabalho busca validar sua proposta, desenvolvendo um sistema que mimetize o funcionamento de um pâncreas humano utilizando lógica difusa e inteligência artificial. O resultado almejado por esta pesquisa, e já alcançado em vários aspectos conforme análise de resultados exposta nessa monografia, é estabilizar a glicose de um paciente, *in silico*, de modo que as métricas clínicas e computacionais sejam devidamente atingidas.

Palavras-chave: Diabetes Tipo 1. Pâncreas Artificial. Lógica Difusa. Inteligência Artificial. Rede Neural.

Abstract

Type 1 Diabetes Mellitus is an autoimmune disease that destroys the insulin-producing beta cells of the pancreas. Therefore, lifelong exogenous hormone replacement becomes essential. However, this is a highly complex process that can lead patients to develop anxiety or depression due to the variety of strategies available for disease control. In this context, closed-loop systems emerge, mimicking the function of the human pancreas. For these systems to function properly, it is necessary to choose an effective strategy for the controller's operation. Among these strategies, fuzzy logic and Artificial Intelligence stand out. Fuzzy logic allows for dealing with the uncertainty and imprecision of physiological data, enabling more flexible decisions. Artificial Intelligence, through a Neural Network, offers the ability to learn and adapt to different patient profiles, which enhances the personalization of treatment. Therefore, through the development of proof-of-concept (PoC) projects, this work seeks to validate its proposal by developing a system that mimics the functioning of a human pancreas using fuzzy logic and artificial intelligence. The desired outcome of this research, already achieved in several aspects as per the results analysis presented in this monograph, is to stabilize a patient's glucose level, *in silico*, so that clinical and computational metrics are properly met.

Keywords: Type 1 Diabetes. Artificial Pancreas. Fuzzy Logic. Artificial Intelligence. Neural Network.

Lista de ilustrações

Figura 1 – Sistema de Malha Aberta.	37
Figura 2 – Sistema de Malha Fechada.	38
Figura 3 – Variável Linguística: Desempenho.	44
Figura 4 – Mecanismo de Inferência Difusa	49
Figura 5 – Sistema de Inferência Difusa	50
Figura 6 – Inferência de Mamdani.	51
Figura 7 – Modelo Neural.	54
Figura 8 – Arquitetura de um Bloco <i>LSTM</i>	60
Figura 9 – Arquitetura de um Bloco <i>GRU</i>	62
Figura 10 – Arquitetura Hexagonal.	67
Figura 11 – BPMN Levantamento Bibliográfico.	77
Figura 12 – BPMN Método Orientado às Provas de Conceito.	80
Figura 13 – Quadro Kanban.	81
Figura 14 – BPMN Método de Análise de Resultados.	83
Figura 15 – BPMN de Atividades e Subprocessos da Primeira Etapa do TCC.	87
Figura 16 – BPMN de Atividades e Subprocessos da Segunda Etapa do TCC.	89
Figura 17 – Sistema de Malha Fechada das POCs.	92
Figura 18 – Diagrama de Classes da Primeira POC.	93
Figura 19 – Diagrama de Classes da Segunda POC.	94
Figura 20 – Função de Pertinência da Variável Linguística <i>BGL</i>	97
Figura 21 – Função de Pertinência da Variável Linguística <i>BGL_RATE</i>	97
Figura 22 – Função de Pertinência da Variável Linguística <i>BGL_ACC</i>	98
Figura 23 – Matriz de Dosagem.	98
Figura 24 – Função de Pertinência da Variável Linguística <i>INSULIN</i>	99
Figura 25 – BPMN Desenvolvimento das Redes Neurais.	101
Figura 26 – Matriz de Correlação - Crianças.	105
Figura 27 – Matriz de Correlação - Adolescentes.	106
Figura 28 – Matriz de Correlação - Adultos.	107
Figura 29 – Gráfico de Glicose da Criança 1.	118
Figura 30 – Gráfico de Glicose do Adolescente 5.	122

Lista de tabelas

Tabela 1 – Resumo das Ferramentas Utilizadas	73
Tabela 2 – Classificação da Pesquisa.	75
Tabela 3 – <i>Strings</i> de Busca Utilizadas.	78
Tabela 4 – Relação entre Fator de Personalização e Fator Escalar	99
Tabela 5 – Classificação dos níveis de risco segundo os valores de HBGI e LBGI .	101
Tabela 6 – Faixa de Valores da Variável Alvo por Faixa Etária	103
Tabela 7 – Quantidade de Passos de Janela Temporal Por Faixa Etária	105
Tabela 8 – Desempenho do modelo por faixa etária	109
Tabela 9 – Agenda Alimentar Padrão por Horário e Cenário com Quantidade de Carboidrato (g).	116
Tabela 10 – Agenda Alimentar Específica por Horário e Cenário com Quantidade de CHO (g).	117
Tabela 11 – Valores de TDD e PF para Diferentes Crianças Simuladas.	117
Tabela 12 – Resultados da Primeira Prova de Conceito: Criança 1.	118
Tabela 13 – Resultados dos Testes de Estresse da Criança 1 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	119
Tabela 14 – Resultados da Primeira Prova de Conceito: Criança 4.	119
Tabela 15 – Resultados dos Testes de Estresse da Criança 4 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	119
Tabela 16 – Resultados da Primeira Prova de Conceito: Criança 7.	119
Tabela 17 – Resultados dos Testes de Estresse da Criança 7 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	120
Tabela 18 – Valores de TDD e PF para Diferentes Adolescentes Simulados.	120
Tabela 19 – Resultados da Primeira Prova de Conceito: Adolescente 2.	120
Tabela 20 – Resultados dos Testes de Estresse do Adolescente 2 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	121
Tabela 21 – Resultados da Primeira Prova de Conceito: Adolescente 5.	121
Tabela 22 – Resultados por Cenário - Adolescente 5	121
Tabela 23 – Resultados dos Testes de Estresse do Adolescente 5 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	122
Tabela 24 – Resultados da Primeira Prova de Conceito: Adolescente 8.	123
Tabela 25 – Resultados dos Testes de Estresse do Adolescente 8 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	123
Tabela 26 – Valores de TDD e PF para Diferentes Adultos Simulados.	123
Tabela 27 – Resultados da Primeira Prova de Conceito: Adulto 3.	123

Tabela 28 – Resultados dos Testes de Estresse do Adulto 3 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	124
Tabela 29 – Resultados da Primeira Prova de Conceito: Adulto 6.	124
Tabela 30 – Resultados dos Testes de Estresse do Adulto 6 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito	125
Tabela 31 – Resultados da Primeira Prova de Conceito: Adulto 9.	125
Tabela 32 – Resultados dos Testes de Estresse do Adulto 9 nos Cenários 5, 6 e 7 . .	125
Tabela 33 – Resultados da Segunda Prova de Conceito: Criança 1.	126
Tabela 34 – Resultados dos Testes de Estresse da Criança 1 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	127
Tabela 35 – Resultados da Segunda Prova de Conceito: Criança 4.	127
Tabela 36 – Resultados dos Testes de Estresse da Criança 4 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	127
Tabela 37 – Resultados da Segunda Prova de Conceito: Criança 7.	128
Tabela 38 – Resultados dos Testes de Estresse da Criança 7 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	128
Tabela 39 – Resultados da Segunda Prova de Conceito: Adolescente 2.	129
Tabela 40 – Resultados dos Testes de Estresse do adolescente 2 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	129
Tabela 41 – Resultados da Segunda Prova de Conceito: Adolescente 5.	129
Tabela 42 – Resultados da Segunda Prova de Conceito: Adolescente 5.	130
Tabela 43 – Resultados dos Testes de Estresse do adolescente 5 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	130
Tabela 44 – Resultados da Segunda Prova de Conceito: Adolescente 8.	130
Tabela 45 – Resultados dos Testes de Estresse do adolescente 8 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	131
Tabela 46 – Resultados da Segunda Prova de Conceito: Adulto 3.	131
Tabela 47 – Resultados dos Testes de Estresse do Adulto 3 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	131
Tabela 48 – Resultados da Segunda Prova de Conceito: Adulto 6.	132
Tabela 49 – Resultados dos Testes de Estresse do Adulto 6 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	132
Tabela 50 – Resultados da Segunda Prova de Conceito: Adulto 9.	133
Tabela 51 – Resultados dos Testes de Estresse do Adulto 9 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito	133
Tabela 52 – Status e Comprobatório dos Objetivos Específicos.	137

Lista de códigos

Código 1 – Código da Classe <i>Glucose_Window</i>	95
Código 2 – Código do Tempo de Medição.	95
Código 3 – Adição das Medições de Glicose.	96
Código 4 – Função para Sugerir Dose.	96
Código 5 – Código da Sugestão de Insulina.	98
Código 6 – Código da Extração de Dados do Simulador.	103

Lista de abreviaturas e siglas

ANN	<i>Artificial Neural Network</i>
CGM	<i>Continuous Glucose Monitoring</i>
CHO	Carboidratos
CNN	<i>Convolutional Neural Network</i>
DDD	<i>Domain-Driven Design</i>
DM1	<i>Diabetes Mellitus</i> Tipo 1
FDA	<i>Food and Drug Administration</i>
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
IDE	<i>Integrated Development Environment</i>
LSTM	<i>Long Short-Term Memory</i>
MPC	<i>Model Predictive Control</i>
PF	<i>Personalization Factor</i>
PID	<i>Proporcional-Integral-Diferencial</i>
PoC	<i>Proof of Concept</i>
RNN	<i>Recurrent Neural Network</i>
RNAs	Redes Neurais Artificiais
TCC	Trabalho de Conclusão de Curso
TDD	<i>Total Dially Dose</i>
TIR	<i>Time in Range</i>
TPU	<i>Tensor Processing Unit</i>

Lista de símbolos

μ	Letra grega minúscula mi
φ	Letra grega minúscula phi
χ	Letra grega minúscula chi
σ	Letra grega minúscula sigma
$\int$	Integral
$\in$	Pertence
$\notin$	Não Pertence
$\iff$	Se e Somente Se
$\forall$	Para Todo
$\cup$	União
$\cap$	Interseção
$\Rightarrow$	Se ... Então
$\wedge$	E
$\vee$	Ou
$ $	Tal que
$\emptyset$	Conjunto vazio
Σ	Somatório

Sumário

1	INTRODUÇÃO	29
1.1	Contextualização	29
1.2	Questionamento	32
1.3	Justificativa	32
1.4	Objetivos	33
1.4.1	Objetivo Geral	33
1.4.2	Objetivos Específicos	33
1.5	Organização da Monografia	34
2	REFERENCIAL TEÓRICO	35
2.1	Sistemas de Controle	35
2.1.1	Definições	35
2.1.1.1	Variável Controlada e Sinal de Controle ou Variável Manipulada	36
2.1.1.2	Plantas	36
2.1.1.3	Processos	36
2.1.1.4	Sistema	36
2.1.1.5	Distúrbios	36
2.1.1.6	Controle com Realimentação	36
2.1.2	Configurações de Sistemas de Controle	37
2.1.2.1	Sistema de Malha Aberta	37
2.1.2.2	Sistema de Malha Fechada	38
2.1.3	Alguns Tipos de Controladores para Sistema de Malha Fechada Aplicados ao Contexto de um Pâncreas Artificial	39
2.1.3.1	<i>Proporcional-Integral-Diferencial Control (PID)</i>	39
2.1.3.2	<i>Model Predictive Control (MPC)</i>	40
2.1.3.3	Controlador de Inteligência Artificial	40
2.1.3.3.1	Controlador por Redes Neurais	41
2.1.3.3.2	Controlador de Lógica Difusa	41
2.2	Lógica Difusa	42
2.2.1	Conjuntos Clássicos	42
2.2.2	Conjuntos Difusos	43
2.2.3	Variáveis Linguísticas	43
2.2.4	Definições e Operações em Conjuntos Difusos	44
2.2.4.1	União	45
2.2.4.2	Interseção	45

2.2.4.3	Complemento	45
2.2.4.4	Propriedades dos Conjuntos Difusos	46
2.2.5	Proposições Difusas	47
2.2.6	Regra de Inferência Composicional	48
2.2.7	Sistemas de Inferência Difusa	49
2.2.7.1	Sistema de Inferência de Mamdani	51
2.3	Aprendizado Profundo	52
2.3.1	Redes Neurais Artificiais (RNAs)	53
2.3.2	Topologia das Redes Neurais	53
2.3.3	Modelos de um Neurônio	54
2.3.4	Tipos de Função de Ativação	54
2.3.4.1	Função Limiar	55
2.3.4.2	Função Linear por Partes	55
2.3.4.3	Função Sigmoides	55
2.3.5	Aprendizagem e Treinamento de RNAs	56
2.3.6	Arquitetura de Redes Neurais	57
2.3.6.1	<i>FeedForward Networks</i> , ou Redes Alimentadas Adiante	57
2.3.6.2	Redes Convolucionais ou CNN	58
2.3.6.3	Redes Recorrentes	59
2.3.6.4	Redes Neurais Autoassociativas ou <i>Autoencoder</i>	59
2.3.7	<i>Long Short-Term Memory (LSTM)</i>	59
2.3.8	<i>Gated Recurrent Unit (GRU)</i>	62
2.3.9	Métricas	63
2.4	<i>Domain Driven Design</i>	65
2.4.1	Novos Conceitos	65
2.5	Arquitetura Hexagonal	66
2.6	Resumo do Capítulo	67
3	SUPORTE TECNOLÓGICO	69
3.1	Ferramentas de Apoio à Pesquisa	69
3.2	Ferramentas de Apoio ao Desenvolvimento	69
3.3	Ferramentas de Apoio à Elaboração da Monografia	72
3.4	Ferramentas de Apoio Complementar	72
3.5	Resumo do Capítulo	72
4	METODOLOGIA	75
4.1	Classificação da Pesquisa	75
4.1.1	Abordagem de Pesquisa	75
4.1.2	Objetivo de Pesquisa	76
4.1.3	Natureza de Pesquisa	76

4.1.4	Procedimentos de Pesquisa	76
4.2	Levantamento Bibliográfico	76
4.2.1	<i>Strings</i> de Busca	77
4.2.2	CrITÉRIOS de Seleção	77
4.2.3	Artigos Seleccionados	78
4.3	Método Orientado a Provas de Conceito	79
4.4	Método de Desenvolvimento	80
4.5	Método de Análise de Resultados	82
4.6	Cronogramas de Atividades	83
4.6.1	Atividades e Subprocessos da Primeira Etapa do TCC	83
4.6.2	Atividades e Subprocessos da Segunda Etapa do TCC	86
4.7	Resumo do Capítulo	88
5	PÂNCREAS ARTIFICIAL	91
5.1	Contextualização	91
5.2	Provas de Conceito	91
5.2.1	Prova de Conceito 1 - Lógica Difusa	94
5.2.1.1	Conjuntos Difusos	96
5.2.1.2	Matriz de Dosagem	97
5.2.1.3	Fator de Personalização	98
5.2.2	Prova de Conceito 2 - Lógica Difusa e Inteligência Artificial	100
5.2.2.1	Sintetização dos Dados	100
5.2.2.2	Tratamento dos Dados	103
5.2.2.3	Treinamento das Redes Neurais	104
5.2.2.3.1	Crianças	106
5.2.2.3.2	Adolescentes	108
5.2.2.3.3	Adultos	108
5.2.2.4	Análise das Métricas	109
5.2.2.5	Utilização do Modelo	110
5.3	Resumo do Capítulo	111
6	ANÁLISE DE RESULTADOS	113
6.1	Plano de Ação	113
6.1.1	Personas e Rotinas	114
6.1.1.1	Testes de Estresse	115
6.2	Coleta de Dados	116
6.2.1	Resultados da Primeira Prova de Conceito	116
6.2.1.1	Crianças	117
6.2.1.2	Adolescentes	120
6.2.1.3	Adultos	122

6.2.2	Resultados da Segunda Prova de Conceito	126
6.2.2.1	Crianças	126
6.2.2.2	Adolescentes	128
6.2.2.3	Adultos	131
6.3	Avaliação do Plano de Execução	133
6.4	Resumo do Capítulo	134
7	CONCLUSÃO	135
7.1	Contexto Geral	135
7.2	Estado Final do Trabalho	136
7.2.1	Questão de Pesquisa	136
7.2.2	Objetivos	136
7.2.2.1	Objetivo Geral	136
7.2.2.2	Objetivos Específicos	137
7.3	Atividades Realizadas	137
7.4	Contribuições	137
7.5	Futuros Trabalhos	138
7.6	Considerações Finais	139
7.7	Resumo do Capítulo	139
	REFERÊNCIAS	141

1 Introdução

Este capítulo tem como intuito acordar o tema deste Trabalho de Conclusão de Curso, bem como a justificativa para sua escolha, delineando o viés da pesquisa e seus principais objetivos. Para isso, será apresentada a [Contextualização](#) do domínio abordado, área da Saúde (mais especificamente, sobre a doença Diabetes), com ênfase na aplicação de lógica difusa em conjunto com algoritmos de aprendizado de máquina para controle de parâmetros associados à doença. Na sequência, será exposta a [Questão de Pesquisa](#) que norteia o desenvolvimento do estudo, acompanhada de sua respectiva Justificativa. Posteriormente, será descrito o [Objetivo Geral e os Objetivos Específicos do trabalho](#), os quais foram cumpridos ao longo da pesquisa. Por fim, será apresentada a [Organização da Monografia](#), detalhando a estrutura em capítulos.

1.1 Contextualização

A Diabetes Mellitus é uma doença crônica caracterizada pela deficiência na produção ou na absorção da insulina, hormônio essencial para a regulação da glicose no sangue e para o fornecimento de energia ao organismo([Ministério da Saúde, 2025](#)). Quando não controlada, essa condição pode levar à hiperglicemia, aumentando significativamente o risco de complicações graves ([EGAN; DINNEEN, 2019](#)) e, em casos extremos, podendo resultar em óbito. Segundo a Organização Mundial da Saúde ([WHO, 2024](#)), cerca de 830 milhões de pessoas em todo o mundo vivem com diabetes, sendo a maioria residente em países de baixa e média renda, onde o acesso ao tratamento é limitado. No Brasil, estima-se que mais de 13 milhões de pessoas convivam com a doença, representando aproximadamente 6,9% da população nacional ([Ministério da Saúde, 2025](#)).

Os principais sintomas da diabetes incluem sede excessiva, necessidade frequente de urinar, visão turva, fadiga e perda de peso involuntária([EGAN; DINNEEN, 2019](#)). A doença pode ser classificada em diferentes tipos, sendo os mais comuns:

- **Diabetes Tipo 1:** Doença crônica de origem autoimune que destrói as células Beta pancreáticas produtoras de insulina, tornando a reposição do hormônio essencial ao tratamento. Representa entre 5% e 10% dos casos no Brasil;
- **Diabetes Tipo 2:** Caracteriza-se pela resistência à insulina e pelo comprometimento da sua produção, sendo frequentemente associada ao sedentarismo, sobrepeso e hipertensão. Representa cerca de 90% dos casos da doença;

- **Diabetes Gestacional:** Surge temporariamente durante a gravidez, elevando os níveis de glicose no sangue sem, contudo, atingir os critérios diagnósticos do diabetes tipo 2, e
- **Outros Tipos Específicos:** Incluem casos de diabetes secundária, associados a doenças pancreáticas, uso de corticosteroides e outras condições médicas.

O diabetes tipo 1, foco deste estudo, resulta de um ataque autoimune às células Beta pancreáticas, levando à produção insuficiente ou inexistente de insulina. Sem esse hormônio, a glicose não consegue ser utilizada pelas células como fonte de energia, acumulando-se no sangue e gerando consequências metabólicas graves. Como resultado, torna-se essencial a reposição exógena do hormônio de maneira vitalícia (ATKINSON; EISENBARTH; MICHELS, 2014). Embora fatores genéticos desempenhem um papel importante, a maioria dos indivíduos diagnosticados com diabetes tipo 1 não apresenta histórico familiar da doença, indicando a influência de fatores ambientais e imunológicos (DIMEGLIO; EVANS-MOLINA; ORAM, 2018).

Ao longo dos anos, os métodos de gerenciamento do diabetes tipo 1 têm evoluído significativamente. Embora esse avanço ocorra, em grande parte, de forma gradual e incremental, ocasionalmente, surgem mudanças mais rápidas e impactantes (DIMEGLIO; EVANS-MOLINA; ORAM, 2018). Um marco essencial foi a descoberta da insulina em 1921-1922 (ATKINSON; EISENBARTH; MICHELS, 2014), seguida pelo desenvolvimento de novos análogos da insulina, com perfis de ação variados, permitindo um controle mais refinado da glicemia (DIMEGLIO; EVANS-MOLINA; ORAM, 2018).

Para um controle eficaz da glicose, é necessário um regime de doses múltiplas que imite a liberação fisiológica da insulina. Isso envolve a administração de insulina basal para controle noturno e entre as refeições, combinada com doses de insulina de ação rápida para cobrir a ingestão de carboidratos e corrigir hiperglicemias (DIMEGLIO; EVANS-MOLINA; ORAM, 2018). Essa insulina pode ser administrada por injeções, utilizando canetas de insulina quando disponíveis, ou por meio de bombas de insulina, que são preferidas por muitos pacientes devido à sua praticidade e ao maior controle (DIMEGLIO; EVANS-MOLINA; ORAM, 2018).

No entanto, um manejo inadequado da insulina pode resultar em complicações graves, como:

- **Hipoglicemia:** Redução excessiva da glicose no sangue, podendo causar sintomas como tremores, sudorese, irritabilidade, confusão mental, convulsões e, em casos mais graves, inconsciência. A hipoglicemia severa requer assistência imediata e pode levar a déficits cognitivos em longo prazo, e

- **Cetoacidose Diabética:** Ocorre quando há um acúmulo de cetonas no sangue devido à deficiência de insulina, podendo resultar em desidratação severa, vômitos, confusão mental e risco de morte.

Para os indivíduos diagnosticados, a diabetes pode representar uma condição vitalícia que exige atenção constante à dieta, ao estilo de vida e ao automonitoramento glicêmico. O tratamento frequentemente envolve medicamentos e dispositivos de monitoramento, que podem ser dolorosos, complexos e onerosos (GÓES; VIEIRA; JÚNIOR, 2007). Essa realidade pode desencadear ansiedade e depressão nos pacientes (EGAN; DINNEEN, 2019).

Perante a necessidade de melhorar o controle de glicose e minimizar as complicações, novas tecnologias têm surgido nos últimos anos, como é o caso dos *CGMs* (*Continuous Glucose Monitors*), dispositivos utilizados para monitoramento contínuo da glicose. Esses dispositivos são minimamente invasivos e possuem precisão semelhante aos monitores capilares de glicemia (DIMEGLIO; EVANS-MOLINA; ORAM, 2018).

Com isso, pode-se aliar os *CGMs* a um algoritmo de controle e uma bomba de insulina. Esses dispositivos assumem uma identidade conhecida como sistema de circuito fechado integrado, resultando em um pâncreas artificial que simula a função do pâncreas ao ajustar automaticamente a liberação de insulina com base nos níveis de glicose monitorados (COBELLI; RENARD; KOVATCHEV, 2011a). Acredita-se que tal sistema facilite um gerenciamento do diabetes de maneira mais segura, simples e eficaz, levando a melhores resultados clínicos (MAUSETH et al., 2010).

De maneira conjunta, a utilização de lógica difusa no domínio da saúde vem crescendo pois, o conhecimento médico é caracterizado por uma "incerteza inescapável". Embora existam muitas razões para essa "incerteza inescapável", a mais importante é a inevitável imprecisão da medicina. A complexidade da prática médica resulta da falta de informação, da imprecisão e da natureza contraditória (GÜRSEL, 2016).

Adicionalmente, os insumos de análise da Diabetes demandam representações usando conjuntos difusos. Na teoria clássica, os elementos podem pertencer ou não ao conjunto. Já em um conjunto difuso, os elementos têm graus de pertinência. Sendo assim, podem pertencer ao conjunto com diferentes níveis de intensidade (ZADEH, 1965). Isso permite tratar de forma mais adequada algumas semânticas inerentes ao contexto da Diabetes, tais como: um dado valor de glicose possui um grau de pertinência de 0.7 ao conjunto de valores que representam "Alta Taxa de Glicose no Sangue"; 0.1 ao conjunto de valores que representam "Baixa Taxa de Glicose no Sangue", dentre outras porcentagens para conjuntos intermediários, compreendidos entre Alta Taxa e Baixa Taxa. Nesse caso, percebe-se que o valor dado é considerado mais alto do que baixo, em termos de taxa de glicose no sangue.

Além do mais, a lógica difusa pode ser utilizada de maneira conjunta com outras técnicas como, por exemplo, redes neurais (GÜRSEL, 2016). Um exemplo de redes neurais são as redes recorrentes nas quais possuem ao menos um laço de realimentação resultando em um comportamento dinâmico não linear e permitindo com que seja analisado pela rede neural o retrospecto da situação (HAYKIN, 2001). Desta maneira, pode-se, utilizar-se das redes neurais recorrentes para auxiliar a lógica difusa na tarefa de estabilizar a glicose de um diabético tipo 1.

Aliado a isso, o desenvolvimento exponencial dos computadores permitiram a realização de simulações através dos mesmos, surgindo então uma nova vertente de estudos denominados *in silico*. Esta vertente é relativamente nova, sendo utilizada inicialmente para prever como certos farmacos interagem com o corpo e com patógenos (MKPB, 2015). Isto permitiu com que os estudos desse trabalho fossem realizados com segurança.

1.2 Questionamento

Ao final deste trabalho, pretende-se responder ao seguinte questionamento:

É possível desenvolver, *in silico*, um sistema de controle de glicose que proporcione maior comodidade ao usuário, minimizando a necessidade de inserção manual de informações pessoais?

Para responder essa questão, utilizou-se, pautado em uma pesquisa prévia, algoritmos de aprendizado de máquina além do uso de lógica difusa. Esse uso será mais bem explicado adiante, no [Capítulo 2 - Referencial Teórico](#) e no [Capítulo 5 - Pâncreas Artificial](#).

1.3 Justificativa

Para ter uma boa qualidade de vida, um melhor metabolismo e a minimização das complicações da diabetes tipo 1, é necessário ter cuidado com o estilo de vida do paciente. Todavia, esse é um processo complicado devido à variedade de estratégias que se pode adotar de acordo com a situação do mesmo. Devido a isso, muitos diabéticos podem acabar desenvolvendo ansiedade ou depressão durante as suas vidas (SIGURÐARDÓTTIR, 2005).

No caso de crianças, o regime de tratamento para diabetes tipo 1 é ainda mais complexo e exigente, demandando verificações frequentes de glicemia; monitoramento cuidadoso de dieta e padrões de exercícios, e múltiplas administrações diárias de insulina. Todos esses cuidados realizados pelos pais ou responsáveis diretos, o que aumenta o estresse dos mesmos devido à preocupação, à ansiedade e às dificuldades no gerenciamento da condição (TSIOULI et al., 2013).

Além disso, pode-se destacar também os riscos em que os pacientes estão constan-

temente expostos durante o tratamento, podendo classificar duas principais complicações: a hipoglicemia e a cetoacidose diabética, ambas potencialmente fatais (EGAN; DINNEEN, 2019).

Ademais, vários modelos de simulação do crescimento de glicose de um diabético tipo 1 foram propostos nos últimos anos, sendo alguns aceitos pela *Food and Drug Administration*. No caso, os aceitos são os que conseguem simular o transporte de glicose subcutânea e tratar os erros eventuais do sensor. Desta forma, substituem os testes *in vivo* para um simulador, de modo que todo o sistema possa ser testado de maneira eficaz previamente (COBELLI; RENARD; KOVATCHEV, 2011b).

Com base nas colocações, faz-se necessária e segura a realização de estudos exploratórios com o propósito de prover uma solução para o gerenciamento da glicose por parte do diabético, de modo que possa mitigar questões como a insegurança constante ao administrar as taxas de glicose; desconhecimento quanto aos comportamentos esperados ao se aplicar as doses recomendadas de insulina, dentre outras, tudo *in silico*.

1.4 Objetivos

Os objetivos deste documento foram divididos entre [Objetivo Geral](#) e [Objetivos Específicos](#).

1.4.1 Objetivo Geral

Como Objetivo Geral, procurou-se cumprir com o desenvolvimento de um controlador, *in silico*, de gerenciamento de glicemia, personalizando e adaptando esse monitoramento com base nas necessidades de cada paciente. Para isso, fez-se uso de lógica difusa para conseguir sugerir uma dose de insulina aproximada ao que o paciente necessita. Na sequência, utilizou-se de Aprendizado de Máquina, mais especificamente Aprendizado Profundo para adequar a dose ao paciente.

1.4.2 Objetivos Específicos

Como objetivos específicos, podem ser destacados:

1. Estudo sobre lógica difusa;
2. Estudo sobre Aprendizado de Máquina e Aprendizado Profundo;
3. Estudo sobre sistemas manuteníveis, os quais detenham a capacidade de atender mudanças contínuas ao longo do seu ciclo de vida;

4. Desenvolvimento de provas de conceito, considerando os conhecimentos adquiridos nos estudos anteriores, visando uma solução minimalista, embasada e testada; e
5. Documentação dos resultados obtidos, de forma a evidenciar a personalização e a adaptação da solução diante de diferentes necessidades dos pacientes, o que confere comodidade aos mesmos.

1.5 Organização da Monografia

Esta monografia está estruturada da seguinte forma:

- **Capítulo 2 - Referencial Teórico:** Exposição do material teórico utilizado como base deste trabalho, com destaque para lógica difusa, redes neurais e sistemas de circuito fechado;
- **Capítulo 3 - Suporte Tecnológico:** Apresentação das principais tecnologias utilizadas neste trabalho;
- **Capítulo 4 - Metodologia:** Exibição do plano metodológico seguido pelo autor para condução do trabalho como um todo;
- **Capítulo 5 - Pâncreas Artificial:** Apresentação detalhada do produto;
- **Capítulo 6 - Resultados Obtidos:** Descrição sobre os resultados obtidos durante o desenvolvimento do produto, e
- **Capítulo 7 - Conclusão:** Descrição sobre os resultados obtidos ao longo da realização do TCC, com destaque para os objetivos cumpridos e questão de pesquisa, bem como a proposição de novas ideias de trabalhos futuros.

2 Referencial Teórico

Neste capítulo, serão percorridos os conceitos fundamentais para a compreensão deste trabalho. A princípio, será discutido sobre o conceito de [Sistemas de Controle](#) e seus devidos comportamentos, representando o domínio de aplicações desse trabalho. Na sequência, serão evidenciados os conceitos fundamentais da [Lógica Difusa](#); [Aprendizado Profundo](#) e suas devidas nuances, bem como *Domain Driven Design* (DDD), e [Arquitetura Hexagonal](#). [Lógica Difusa](#) e [Aprendizado Profundo](#) são usados como insumos para viabilizar a solução, sendo essa um simulador orientado à Inteligência Artificial. *Domain Driven Design* é importante para condução do desenvolvimento dessa solução centrando-se em um modelo de domínio, que atende processos e princípios específicos. Já a [Arquitetura Hexagonal](#) é a arquitetura de software escolhida para estruturar os componentes arquiteturais dessa solução, considerando ser uma abordagem agnóstica em termos de tecnologias; permitindo encapsulamento da lógica em um núcleo, bem como testabilidade do software. Por fim, será apresentado o [Resumo do Capítulo](#).

2.1 Sistemas de Controle

Um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de se obter uma saída com um desempenho desejado, dada uma entrada especificada ([NISE, 2023](#)).

De acordo com [Nise \(2023\)](#), através dos sistemas de controle, pode-se extrair as seguintes vantagens:

1. **Amplificação de potência:** Produzir grande ganho de potência necessário para um determinado fim;
2. **Controle remoto:** Operar robôs para operações perigosas;
3. **Conveniência de forma de entrada:** Alterar saídas conforme entradas, e
4. **Compensação de perturbações:** Realizar o controle de determinadas variáveis, de forma que o sistema seja capaz de retornar uma saída correta de acordo com a perturbação observada.

2.1.1 Definições

De acordo com [Ogata \(2010\)](#), pode-se definir as seguintes terminologias específicas, conforme acordado nas próximas seções.

2.1.1.1 Variável Controlada e Sinal de Controle ou Variável Manipulada

A variável controlada é a grandeza ou a condição que é medida e controlada. O sinal de controle ou variável manipulada é a grandeza ou a condição modificada pelo controlador, de modo que afete o valor da variável controlada. Normalmente, a variável controlada é a saída do sistema.

2.1.1.2 Plantas

Uma planta pode ser uma parte de equipamento ou apenas um conjunto de componentes de um equipamento que funcione de maneira integrada, com o objetivo de realizar determinada operação. De maneira mais sucinta, é qualquer objeto físico a ser controlado.

2.1.1.3 Processos

Um processo é uma operação natural de progresso contínuo; ou um desenvolvimento caracterizado por uma série de modificações graduais que se sucedem umas às outras de modo relativamente estável, avançando em direção a um dado resultado ou objetivo. Também pode ser uma operação contínua progressiva, artificial ou voluntária, que consiste em uma série de ações ou movimentos controlados, sistematicamente destinados a atingir determinados fins ou resultados. De maneira mais direta, pode ser toda operação a ser controlada.

2.1.1.4 Sistema

Um sistema é a combinação de componentes que agem em conjunto para atingir determinado objetivo. A ideia de sistema não fica restrita apenas a algo físico, podendo envolver sistemas abstratos dinâmicos, como sistemas físicos, biológicos e econômicos.

2.1.1.5 Distúrbios

Um distúrbio é um sinal que tende a afetar de maneira adversa o valor da variável de saída de um sistema. Se um distúrbio for gerado dentro de um sistema, ele será chamado de distúrbio interno; Enquanto um distúrbio externo é aquele gerado fora do sistema e que se comporta como um sinal de entrada no sistema.

2.1.1.6 Controle com Realimentação

Refere-se à uma operação que, na presença de distúrbios, tende a diminuir a diferença entre a saída de um sistema e alguma entrada de referência, atuando com base nessa diferença.

2.1.2 Configurações de Sistemas de Controle

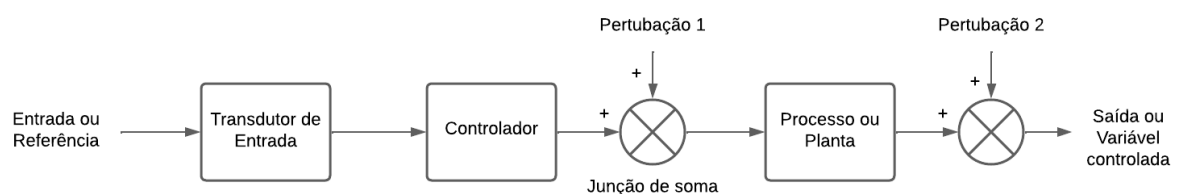
Os sistemas de controle possuem duas principais configurações: malha aberta e malha fechada. Essas configurações podem ser consideradas como a arquitetura interna do sistema total (NISE, 2023).

2.1.2.1 Sistema de Malha Aberta

Os chamados sistemas de controle de malha aberta são aqueles em que o sinal de saída não exerce ação alguma de controle no sistema. Isso quer dizer que, em um sistema de controle de malha aberta, o sinal de saída não é medido, nem realimentado, para comparação com a entrada (OGATA, 2010).

A sua arquitetura é demonstrada de maneira genérica pela Figura 1. No geral, ele começa com um subsistema chamado de transdutor de entrada, o qual converte a forma da entrada para aquela utilizada pelo controlador. O controlador aciona um processo ou uma planta. A entrada algumas vezes é chamada de referência; enquanto a saída pode ser chamada de variável controlada. Outros sinais, como as perturbações, são mostrados adicionados às saídas do controlador e do processo através de junções de soma, as quais fornecem a soma algébrica dos seus sinais de entrada utilizando os sinais associados (NISE, 2023).

Figura 1 – Sistema de Malha Aberta



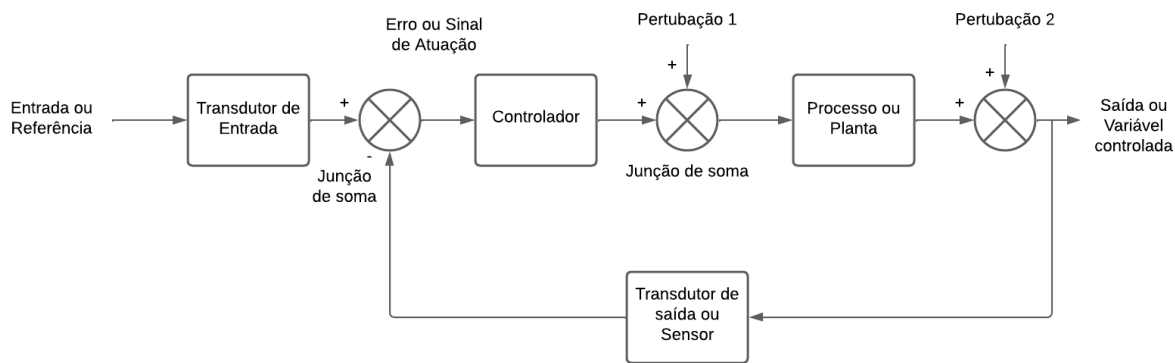
Fonte: (NISE, 2023).

Em qualquer sistema de controle de malha aberta, a saída não é comparada com a entrada de referência. Assim, cada entrada de referência corresponde à uma condição fixa de operação. Dessa maneira, a precisão do sistema depende de uma calibração. Na presença de distúrbios, um sistema de controle de malha aberta não vai executar a tarefa desejada. Na prática, o sistema de controle de malha aberta somente poderá ser utilizado caso a relação entre a entrada e a saída for conhecida, e não ocorrendo distúrbio algum interno ou externo(OGATA, 2010).

2.1.2.2 Sistema de Malha Fechada

Os sistemas de malha fechada, também chamados de sistemas de controle com realimentação, são sistemas que estabelecem uma relação de comparação entre a saída e a entrada de referência, utilizando a diferença como meio de controle. Em um sistema de controle de malha fechada, o sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal de realimentação, realimenta o controlador, de modo a minimizar o erro e acertar a saída do sistema ao valor desejado. O termo ‘controle de malha fechada’ sempre implica na utilização do controle com realimentação para reduzir o erro do sistema(OGATA, 2010). A sua arquitetura genérica é demonstrada pela Figura 2:

Figura 2 – Sistema de Malha Fechada



Fonte: (NISE, 2023).

Segundo Nise (2023), o transdutor de entrada converte a forma da entrada para a forma utilizada pelo controlador. Um transdutor de saída, ou sensor, mede a resposta da saída e a converte para a forma utilizada pelo controlador. A primeira junção de soma adiciona algebricamente o sinal de entrada ao sinal de saída, que chega através da malha de realimentação, o caminho de retorno da saída para a junção de soma.

Na Figura 2, o sinal de saída é subtraído do sinal de entrada. O resultado, geralmente, é chamado de sinal de atuação. Entretanto, nos sistemas em que ambos os transdutores, de entrada e da saída, possuem ganho unitário (isto é, o transdutor amplifica sua entrada por um fator igual a 1), o valor do sinal de atuação é igual à diferença real entre a entrada e a saída. Nessas condições, o sinal de atuação é chamado de erro.

Desta maneira, o sistema em malha fechada compensa o efeito das perturbações medindo a resposta da saída, realimentando essa medida através da malha de realimentação e comparando essa resposta com a entrada na junção de soma. Se existir qualquer diferença entre as duas respostas, o sistema aciona a planta, através do sinal de atuação, para fazer uma correção. Não havendo, o sistema não aciona a planta, uma vez que a resposta da planta já é a resposta desejada(NISE, 2023).

2.1.3 Alguns Tipos de Controladores para Sistema de Malha Fechada Aplicados ao Contexto de um Pâncreas Artificial

No contexto de um pâncreas artificial, alguns tipos de controladores de sistemas de malha fechada destacam-se: *Proporcional-Integral-Diferencial* (PID), Controle de Predição de Modelo (MPC), e Controle por Inteligência Artificial (MING et al., 2024).

2.1.3.1 *Proporcional-Integral-Diferencial Control* (PID)

Utilizados há muito tempo para casos industriais, em 1989, os PIDs representavam cerca de 90% dos algoritmos de controladores utilizados. Sua difusão pode ser atribuída à simplicidade e à facilidade de ajustes em funcionamento (BANSAL; SHARMA; SHREERAMAN, 2012). Tal algoritmo possui alta confiabilidade, bem como uma robustez excelente (MING et al., 2024). Além de ser utilizado em máquinas industriais, pode-se utilizar um controlador PID para tarefas de otimização, inteligência artificial, algoritmos genéticos, simulações, dentre outros (BANSAL; SHARMA; SHREERAMAN, 2012).

A sua saída representa a soma de três termos: proporcional, integral e derivativo, sendo cada um deles dependente do valor de erro entre a entrada e a saída (BANSAL; SHARMA; SHREERAMAN, 2012). A Equação 2.1, descrita por Ming et al. (2024), representa a saída de um controlador PID:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t); \quad (2.1)$$

Onde K_p , K_i e K_d são os parâmetros proporcionais, integrais e derivativos, respectivamente. K_i e K_d podem ser escritos como apresentado na Equação 2.2:

$$K_i = \frac{K_p}{T_i}, \quad K_d = K_p T_d; \quad (2.2)$$

Onde T_i e T_d são o tempo de *reset* e o tempo derivativo, respectivamente.

Cada um desses termos possui o seu devido papel descrito por Ming et al. (2024):

- **Proporcional:** acelera a resposta do sistema. Além disso, minimiza, mas não elimina o erro de estado estacionário. Em um pâncreas artificial, ele é representado pela quantidade de insulina secretada quando a medição de glicose desvia do valor alvo, ajustando assim a quantidade de insulina secretada;
- **Derivativo:** reduz a resposta oscilatória do sistema. Em um pâncreas artificial, corresponde à regulação rápida da secreção de insulina quando a glicose muda rapidamente, e

- **Integral:** aumenta a velocidade de resposta do sistema, ao custo de oscilações sustentadas.

Desta maneira, a Equação 2.3 representa a saída de um controlador PID para um pâncreas artificial:

$$PID(t) = K_p(G_b - G) + K_i \int (G_b - G)dt + K_d \frac{d}{dt}(G_b - G); \quad (2.3)$$

Onde $PID(t)$ representa a insulina secretada pelas células beta; G_b é a glicose alvo; G é a medida da glicose em tempo real, e K_p , K_i e K_d são os ganhos dos termos proporcionais, integrais e diferenciais, respectivamente (MING et al., 2024).

Para um pâncreas artificial, os controladores PIDs e suas variações ainda carecem de experimentação, além de apresentarem uma acurácia limitada e resistência a perturbações externas (MING et al., 2024).

2.1.3.2 Model Predictive Control (MPC)

O objetivo geral do *design* do controle preditivo do modelo é calcular uma trajetória de uma variável manipulada futura u para otimizar o comportamento futuro da saída da planta y . A otimização é realizada dentro de uma janela de tempo limitada, fornecendo informações da planta no início da janela de tempo (WANG, 2009). Além disso, ele pode ser facilmente customizado de acordo com as necessidades fisiológicas do paciente, coletando dados passados e combinando-os para formar futuras *inputs* para o sistema, proporcionando assim uma saída mais precisa e adaptada às condições do paciente (MING et al., 2024).

O MPC utiliza um modelo interno da planta para prever saídas futuras e ajustar os comandos de controle, reduzindo o erro em relação à trajetória de referência. Ele pode lidar com múltiplas entradas e saídas, atrasos no sistema e restrições explícitas, além de possuir robustez contra erros de modelagem (MATLAB, 2025).

Assim como o PID, o MPC e suas variações apresentam uma certa carência experimental. Desta forma, há vários modelos que não foram verificados em pacientes diabéticos de todas as faixas etárias (MING et al., 2024).

2.1.3.3 Controlador de Inteligência Artificial

Os controladores através de Inteligência Artificial provêm personalização conforme a demanda da dosagem de insulina. Ao combinar com um Continuous Glucose Monitoring (CGM), podem prever hipoglicemias e tomar uma atitude para preveni-las (MING et al., 2024). Neste caso, existem dois controladores que se destacam: Controlador por Redes Neurais e Controlador de Lógica Difusa.

2.1.3.3.1 Controlador por Redes Neurais

Recentemente, vários pesquisadores estão utilizando redes neurais para controle de processos não lineares. Isso decorre de sua capacidade teórica de aproximar mapeamentos não lineares arbitrários. Redes também podem atingir modelagem mais parcimoniosa do que esquemas de aproximação alternativos (HUNT et al., 1992), como é o caso de um pâncreas artificial. A capacidade de adaptação de uma rede neural permitiu que as taxas de hipoglicemia caíssem até metade do que eram antes de sua aplicação(MING et al., 2024).

Tal controlador aumenta sua velocidade de cálculo conforme sua tarefa. Todavia, apresenta problemas de *overfitting*, os quais podem ser eliminados ao utilizar-se da sua combinação com outros algoritmos. Métodos de controle em malha fechada e estratégias de terapia baseadas em controle ótimo não linear demonstraram eficácia na prevenção de hipoglicemia(MING et al., 2024).

2.1.3.3.2 Controlador de Lógica Difusa

Um controlador de lógica difusa auxilia o processamento de grande quantidade de dados com a ideia de "pertencimento parcial" a um conjunto. No contexto de um pâncreas artificial, ele resume a experiência acumulada ao longo do tratamento da diabetes. Esse modelo é mais fácil de integrar com hardware e não necessita de muita complexidade. Todavia, ele não pode ser mudado após o início do sistema (MING et al., 2024).

A combinação de lógica difusa e Artificial Neural Network (ANN) mostrou melhorias no ajuste da infusão de glicose. Sistemas *fuzzy-ANN* baseados em regras de inferência apresentaram bom desempenho, mas exigiram ajustes manuais para cada paciente (MING et al., 2024).

Estudos sugerem que o controlador de lógica difusa é um método seguro para automatizar a regulação de glicemia em indivíduos sedentários, conseguindo abordar mudanças fisiológicas como ciclo menstrual, doenças e exercícios (MAUSETH et al., 2013). Além disso, algumas variações permitem flexibilidade utilizando o fator de personalização, garantindo melhor qualidade de vida e diminuindo a carga de trabalho para a pessoa diabética (MAUSETH et al., 2010).

Para medir a eficácia do pâncreas artificial desenvolvido neste trabalho, foram adotadas as seguintes métricas de avaliação da qualidade de cada controlador nos diferentes cenários:

- **Tempo Dentro da Faixa-Alvo (TIR):** Em pacientes com DM1 não-gestantes, é recomendado que o tempo na meta glicêmico (Time in Range - TIR) esteja acima

de 70%, com o limite definido por glicemias entre 70 e 180 mg/dL (PITITTO et al., 2023);

- **Tempo Abaixo da Faixa-Alvo:** Em pacientes com DM1 não-gestantes, é recomendado que o tempo em hipoglicemia nível 1 (glicemia < 70 mg/dL) seja menor do que 4%; e o tempo em hipoglicemia nível 2 (glicemia < 54 mg/dL) seja menor que 1%, obtidos através de dispositivos de monitorização contínua de glicose (PITITTO et al., 2023), e
- **Coeficiente de Variação:** Em pacientes não-gestantes com DM1, é recomendado que a variabilidade glicêmica, medida através do cálculo do coeficiente de variação deva ser $< 36\%$ (PITITTO et al., 2023), segundo Lazar et al. (2024) o calculo dessa métrica se dá pela divisão do desvio padrão das glicoses pelo valor médio das mesmas.

Vale ressaltar que, no caso de um diabético tipo 1, para o tratamento sutir efeito, não basta apenas um método para controlar a glicose, como o uso de insulina, é necessário manter uma alimentação saudável e adequada (SAÚDE, 2022).

2.2 Lógica Difusa

Um conjunto difuso é uma classe de objetos com um *continuum* de graus de pertinência. Tal conjunto é caracterizado por uma função de pertinência que atribui a cada objeto uma grandeza de pertinência variando entre zero e um (ZADEH, 1965). A lógica difusa foi concebida como um meio de manipulação e processamento de informações vagas em um universo de incertezas. Essa abordagem tem-se desenvolvido para prover algoritmos de processamento suave da informação, os quais raciocinam e utilizam dados imprecisos. Ao permitir participação parcial dos elementos de um conjunto, transições suaves de uma regra para outra são possíveis. Tal propriedade é desejável para modelagem e controle de sistemas (NAGAMINE, 2001).

2.2.1 Conjuntos Clássicos

Segundo Klir e Yuan (1995), na teoria clássica dos conjuntos, um elemento pode pertencer ou não a um determinado conjunto. Para definir essa relação, utiliza-se a chamada função característica, que determina se um elemento x de um conjunto X pertence a um conjunto S . Dessa forma, para conjuntos clássicos, tem-se a Equação 2.4:

$$\chi_A(X) = \begin{cases} 1, & \text{se } x \in S \\ 0, & \text{se } x \notin S \end{cases} \quad (2.4)$$

Assim, ocorre um mapeamento dos elementos de um conjunto X para outro conjunto composto apenas pelos valores 0 e 1, formalmente expresso com base na relação:

$$\chi_A : X \rightarrow \{0, 1\};$$

2.2.2 Conjuntos Difusos

Na lógica difusa, proposta por Zadeh (1965), também chamada de lógica nebulosa, existe o conceito de pertencimento parcial a um conjunto. Para isso, é necessário definir uma função característica que descreva o grau de pertinência de um elemento x a um conjunto A , atribuindo-lhe um valor dentro do intervalo $[0, 1]$ (CHEN; PHAM, 2000). Assim, um subconjunto difuso A de um universo U é caracterizado por sua função de pertinência:

$$\mu_A : U \rightarrow [0, 1];$$

A título de exemplo, define-se o conjunto difuso F pertencente ao conjunto dos números naturais, conforme a Equação 2.5:

$$F = \{x \in R \mid x \text{ é próximo de } 2\}; \quad (2.5)$$

Define-se a função de pertinência com base na proximidade de um valor real ao ponto 2, conforme a Equação 2.6:

$$\mu_F(x) = \begin{cases} 1 - |x - 2|, & \text{se } 1 < x < 3 \\ 0, & \text{se } x \notin [1, 3] \end{cases} \quad (2.6)$$

Dessa forma, tem-se que $\mu_F(2,001) = 0,999$ e $\mu_F(7) = 0$, ou seja, 7 não é próximo de 2; enquanto 2,001 é muito próximo (SILVA, 2011). A lógica difusa, portanto, oferece uma base para um raciocínio aproximado por meio de proposições imprecisas, utilizando a teoria dos conjuntos difusos (CHEN; PHAM, 2000).

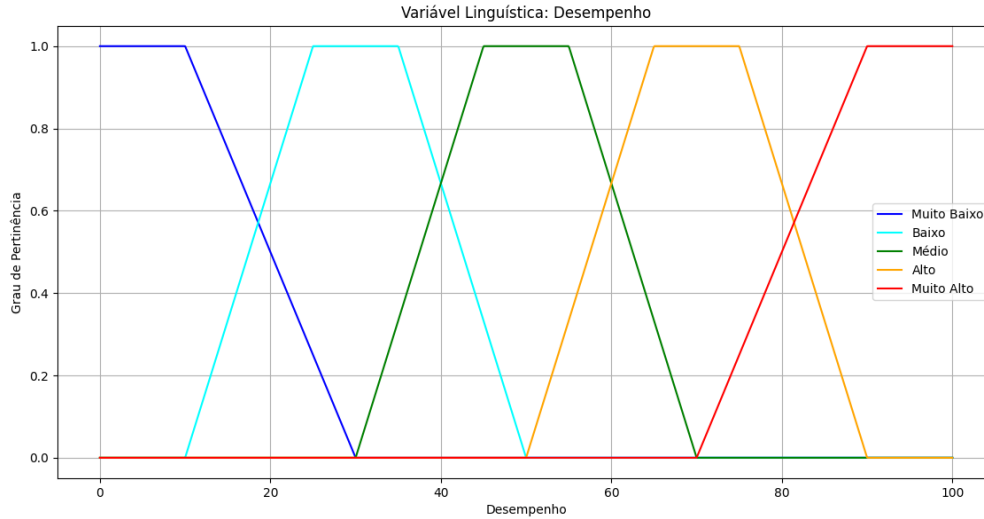
2.2.3 Variáveis Linguísticas

O conceito de um número difuso desempenha um papel fundamental na formulação de variáveis difusas quantitativas. Quando os números difusos representam conceitos linguísticos, como muito pequeno, pequeno, médio e assim por diante, conforme interpretado em um contexto particular, as construções resultantes são geralmente chamadas de variáveis linguísticas (KLIR; YUAN, 1995).

Cada variável linguística representa um conjunto de estados descritos por termos linguísticos. Para cada termo de entrada, deve-se associar um conjunto difuso por meio de uma função de pertinência (KLIR; YUAN, 1995).

Por exemplo, a Figura 3 apresenta a variável linguística desempenho.

Figura 3 – Variável Linguística: Desempenho



Fonte: Autor.

Tal variável assume cinco valores básicos: muito baixo, baixo, médio, alto e muito alto. Cada termo linguístico possui uma função de pertinência trapezoidal dentro do intervalo $[0, 100]$, representando uma restrição difusa nesse domínio (KLIR; YUAN, 1995).

2.2.4 Definições e Operações em Conjuntos Difusos

A exemplo de como ocorre nos conjuntos ordinários, há uma série de definições e operações envolvendo conjuntos difusos (TANSCHKEIT, 2004). Um conjunto difuso A em X é vazio, se e somente se, sua função de pertinência é igual a zero sobre todo U conforme a Equação 2.7:

$$A = \emptyset \iff \forall x \in U, \quad \mu_A(x) = 0; \quad (2.7)$$

De maneira análoga, A e B são conjuntos iguais, se suas funções de pertinência forem iguais sobre todo U conforme a Equação 2.8 :

$$A = B \iff \forall x \in U, \quad \mu_A(x) = \mu_B(x); \quad (2.8)$$

De acordo com [Chen e Pham \(2000\)](#), assim como nos conjuntos clássicos, os conjuntos difusos possuem operações típicas, tais como: união, interseção e complementação. Tem-se três conjuntos difusos A , B e C de um universo U , com funções de pertinência μ_A , μ_B e μ_C , respectivamente.

2.2.4.1 União

A união de dois conjuntos difusos A e B é um novo conjunto difuso, cujo grau de pertinência é determinado pelo maior valor das funções de pertinência dos conjuntos originais, como ocorre na Equação 2.9:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad x \in U; \quad (2.9)$$

Vale ressaltar sobre a propriedade associativa da união entre dois conjuntos ([ZADEH, 1965](#)), conforme a Equação 2.10:

$$\max\{\mu_A(x), \max\{\mu_B(x), \mu_C(x)\}\} = \max\{\max\{\mu_A(x), \mu_B(x)\}, \mu_C(x)\}; \quad (2.10)$$

Uma maneira mais intuitiva de se compreender é considerar que a união entre dois conjuntos difusos A e B é o menor subconjunto contendo ambos ([ZADEH, 1965](#)). Segundo [Zadeh \(1965\)](#), a união entre dois conjuntos difusos A e B também pode ser descrita por:

$$A \cup B;$$

2.2.4.2 Interseção

A interseção entre dois conjuntos difusos A e B é definida pelo menor valor das funções de pertinência, dado pela Equação 2.11:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad x \in U; \quad (2.11)$$

Analogamente à união, a interseção de dois conjuntos é o maior conjunto difuso que contém os conjuntos operandos ([ZADEH, 1965](#)). Segundo [Zadeh\(1965\)](#), a interseção entre dois conjuntos difusos A e B também pode ser descrita por:

$$A \cap B$$

2.2.4.3 Complemento

Segundo [Zadeh \(1965\)](#), o complemento de um conjunto difuso A é definido pela Equação 2.21:

$$\mu_{A'}(x) = 1 - \mu_A(x); \quad (2.12)$$

2.2.4.4 Propriedades dos Conjuntos Difusos

De acordo com Tanscheit (2004), utilizando os operadores *max* e *min* para a descrição da união e interseção difusa, respectivamente, é fácil de verificar as seguintes propriedades algébricas nos conjuntos difusos, representadas pelas Equações 2.13 à 2.20

- **Involução:**

$$(A')' = A; \quad (2.13)$$

- **Idempotência:**

$$\begin{cases} A \cap A = A \\ A \cup A = A \end{cases} \quad (2.14)$$

- **Comutatividade:**

$$\begin{cases} A \cap B = B \cap A \\ A \cup B = B \cup A \end{cases} \quad (2.15)$$

- **Associatividade:**

$$\begin{cases} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cup B) \cup C = A \cup (B \cup C) \end{cases} \quad (2.16)$$

- **Distributividade:**

$$\begin{cases} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{cases} \quad (2.17)$$

- **Absorção:**

$$\begin{cases} A \cap (A \cup B) = A \\ A \cup (A \cap B) = A \end{cases} \quad (2.18)$$

- **Lei Transitiva:**

$$\text{se } A \subset B \text{ e } B \subset C \Rightarrow A \subset C; \quad (2.19)$$

- **Leis de De Morgan:**

$$\begin{cases} (A \cap B)' = A' \cup B' \\ (A \cup B)' = A' \cap B' \end{cases} \quad (2.20)$$

2.2.5 Proposições Difusas

Segundo [Gomide, Gudwin e Tanscheit \(1995\)](#), seja uma frase da forma $(\Pi \text{ é } A)$, onde Π é o nome de uma variável linguística, e A é um subconjunto difuso definido no universo de discurso U de Π . A frase $(\Pi \text{ é } A)$ é chamada de uma proposição difusa. De maneira mais genérica, pode-se ter uma proposição difusa n -ária, efetuando o produto cartesiano de variáveis linguísticas e utilizando relações difusas ao invés de conjuntos difusos.

Proposições difusas podem ser combinadas utilizando-se diferentes operadores, gerando novas proposições difusas. Exemplos de operadores incluem os conectivos lógicos E e OU, bem como o operador de implicação SE...ENTÃO. As proposições difusas resultantes da combinação podem ser descritas em termos de relações difusas. A determinação do valor desta relação difusa, em função dos conjuntos difusos de cada operando, pode ser realizada de muitas maneiras diferentes.

Por exemplo, sejam as variáveis linguísticas de nomes x e y com universos de discurso X e Y , respectivamente. Sejam A e B conjuntos difusos definidos, respectivamente, em X e Y . Por fim, sejam as seguintes proposições difusas $(x \text{ é } A)$ e $(y \text{ é } B)$.

Conectando-se as duas proposições com o operador OU, tem-se:

$$(x \text{ é } A) \vee (y \text{ é } B)$$

Essa combinação gerará a proposição difusa binária:

$$((x, y) \text{ é } R_{A \vee B})$$

Agora, ao conectar as proposições com o operador E, tem-se:

$$(x \text{ é } A) \wedge (y \text{ é } B)$$

que gerará a proposição difusa n -ária:

$$((x, y) \text{ é } R_{A \wedge B})$$

Da mesma forma, segundo [Tanscheit \(2004\)](#), o operador SE ... ENTÃO é também conhecido como declaração condicional difusa, e descreve a dependência do valor de uma variável linguística em relação ao valor de outra. Em muitas aplicações, essas declarações condicionais são simplesmente denominadas regras linguísticas, constituindo-se em frases da forma:

Se x é A então y é B .

Em muitas aplicações, essas declarações condicionais são simplesmente denominadas regras de implicação, sendo representadas por uma relação $R_{A \rightarrow B}$, expressa pela função de pertinência, dada pela Equação 2.21:

$$\mu_{A \rightarrow B}(x, y) = f_{\rightarrow}(\mu_A(x), \mu_B(y)), \quad (2.21)$$

onde $f_{\rightarrow}$ é o operador de implicação.

Quando uma declaração condicional apresenta mais do que uma variável antecedente (x e A), as diversas variáveis são geralmente combinadas por meio do conectivo **e**:

Se $(x_1 \text{ é } A_1)$ e $(x_2 \text{ é } A_2)$ e $\dots$ e $(x_m \text{ é } A_m)$ então $(y \text{ é } B)$,

Várias declarações podem ser combinadas por meio do conectivo **ou**, gerando um conjunto de regras difusas R_N .

R^1 : se $(x \text{ é } A^1)$ então $(y \text{ é } B^1)$ ou

R^2 : se $(x \text{ é } A^2)$ então $(y \text{ é } B^2)$ ou

$\vdots$

R^n : se $(x \text{ é } A^n)$ então $(y \text{ é } B^n)$

Vale ressaltar, que as variáveis Antecedentes são aquelas que vem acompanhado do SE, já aquelas Consequentes, são aquelas acompanhadas do ENTÃO.

2.2.6 Regra de Inferência Composicional

Em conjuntos não difusos, sendo P e Q duas relações não difusas, pode-se realizar a composição de conjuntos através de dois métodos (TANSCHKEIT, 2004):

- **Composição max-min** vide Equação 2.22:

$$f_R(x, z) = f_{P \circ Q}(x, z) = \{(x, z), \max_y[\min(f_P(x, y), f_Q(y, z))]\}; \quad (2.22)$$

- **Composição max-produto** vide Equação 2.23:

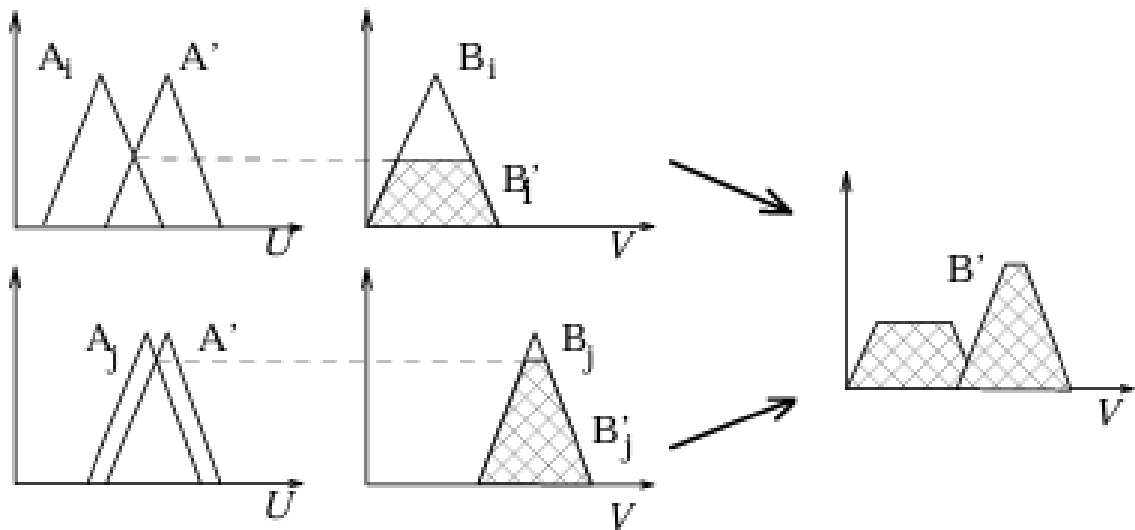
$$f_R(x, z) = f_{P \circ Q}(x, z) = \{(x, z), \max_y[f_P(x, y) \cdot f_Q(y, z)]\}; \quad (2.23)$$

Para realizar a composição de duas variáveis difusas, pode-se utilizar da regra de inferência *max-min* proposta por Zadeh em 1973. No geral, essa regra é descrita pela equação 2.24

$$\mu_B = \sup \{ \mu_A(x, y) * \mu_Q(y, z) \} \quad (2.24)$$

Onde * pode ser substituído ou pelo método do mínimo ou pelo produto. Quando mais de uma regra é acionada, as contribuições das diversas regras após a inferência são combinadas pelo operador de agregação (GOMIDE; GUDWIN; TANSCHKEIT, 1995). Desta maneira, a Figura 4 descreve o processo de inferência *max-min* quando existem duas regras, $A_i \rightarrow B_i$ e $A_j \rightarrow B_j$. A' é o fato de entrada, representado como um conjunto difuso:

Figura 4 – Mecanismo de Inferência Difusa



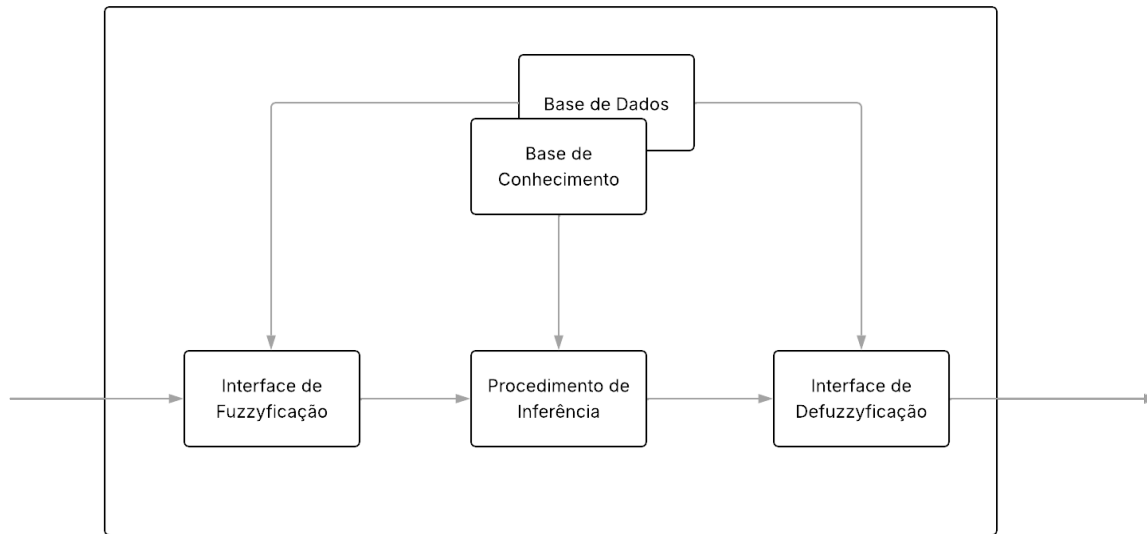
Fonte: (GOMIDE; GUDWIN; TANSCHKEIT, 1995).

2.2.7 Sistemas de Inferência Difusa

A idéia básica em controle difuso é modelar as ações a partir de conhecimento especialista, ao invés de, necessariamente, modelar o processo em si (GOMIDE; GUDWIN, 1994). A Figura 5 revela a estrutura de um sistema de inferência difusa aplicado a um controlador.

Em um sistema de inferência Difusa consideram-se entradas precisas resultantes de medições ou observações. Em virtude disto, é necessário efetuar-se um mapeamento destes dados precisos para os conjuntos difusos (de entrada) relevantes, o que é realizado na interface de *fuzzyficação* (TANSCHKEIT, 2004). Neste estágio, são tomados os valores

Figura 5 – Sistema de Inferência Difusa



Fonte: Autor.

das variáveis de entrada, *fuzzificando-as* de modo que as mesmas se tornem instâncias das variáveis linguísticas(GOMIDE; GUDWIN, 1994).

Por conseguinte, tem-se a base de conhecimento que consiste de uma base de regras, caracterizando a estratégia de controle e suas metas.O sistema só terá bom desempenho, se as regras que o estabelecem forem consistentes(TANSCHAIT, 2004).Observa-se a base de dados, na qual armazenam-se as definições necessárias sobre discretizações e normalizações dos universos de discurso, as partições fuzzy dos espaços de entrada e saída e as definições das funções de pertinência(GOMIDE; GUDWIN, 1994).

Em conjunto com a base de conhecimento, apresenta-se o procedimento de inferência, qual se processam os dados difusos de entrada, junto com as regras, de modo a inferir as ações de controle difuso, aplicando o operador de implicação difuso e as regras de inferência da lógica difusa (GOMIDE; GUDWIN, 1994). Nele, ocorrem as operações com os conjuntos difusos propriamente ditos (TANSCHAIT, 2004).

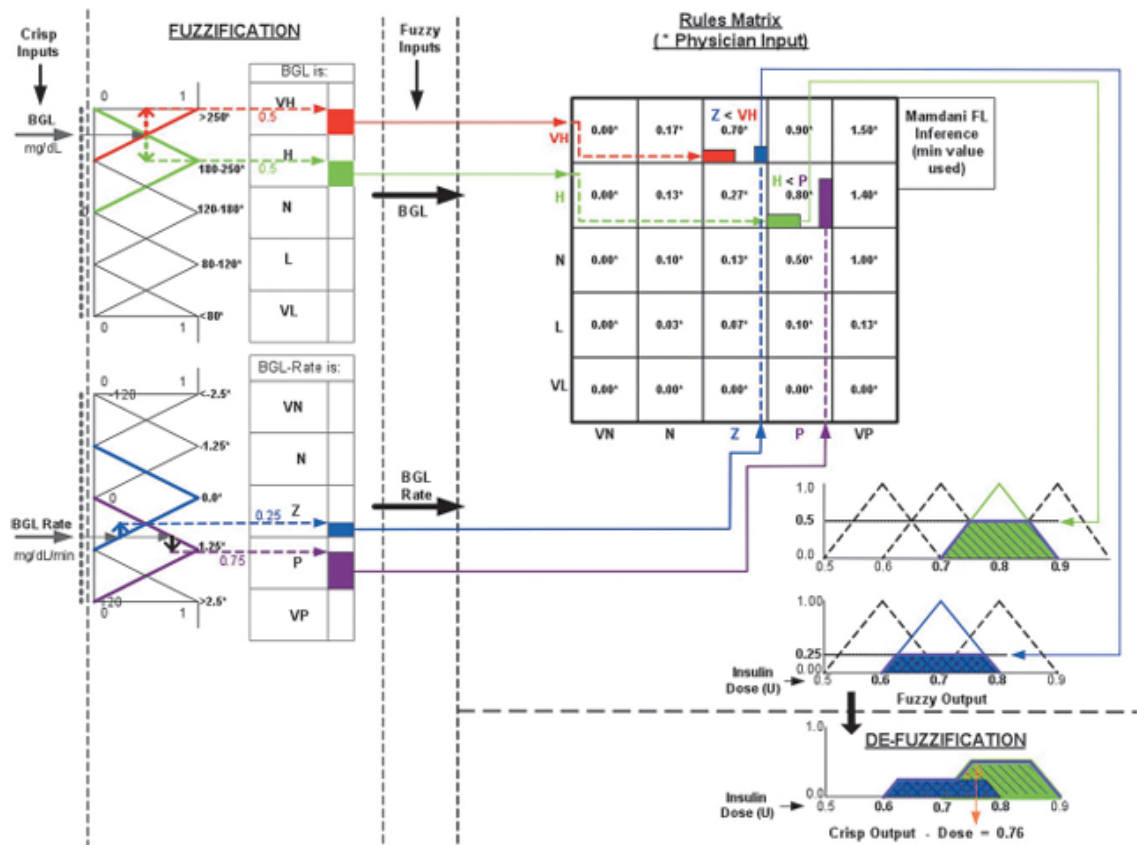
Por fim, haverá a interface de *defuzzyficação* na qual se transforma as ações de controle difuso inferidas em ações de controle não-fuzzy(GOMIDE; GUDWIN, 1994). Existem vários métodos de *defuzzyficação* na literatura; dois dos mais empregados são o centro de gravidade e a média dos máximos. Neste último, a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de pertinência do consequente. Com o centro de gravidade, a saída é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais(TANSCHAIT, 2004).

Um aspecto importante é a definição dos conjuntos difusos correspondentes às variáveis de entrada (antecedentes) e à(s) de saída (consequente(s)), pois o desempenho do sistema de inferência dependerá do número de conjuntos e de sua forma. Pode-se efetuar uma sintonia "manual" das funções de pertinência dos conjuntos, mas é mais comum empregarem-se métodos automáticos. A integração entre sistemas de inferência difusos e redes neurais originando os sistemas *neuro-fuzzy* – ou algoritmos genéticos tem se mostrado adequada para a sintonia de funções de pertinência, assim como para a geração automática de regras (TANSCHKEIT, 2004).

2.2.7.1 Sistema de Inferência de Mamdani

A técnica de inferência difusa mais comumente usada é o chamado método Mamdani, que foi proposto por Mamdani e Assilian como a primeira tentativa de controlar uma combinação de máquina a vapor e caldeira sintetizando um conjunto de regras de controle linguístico obtidas de operadores humanos experientes (IANCU, 2012). A Figura 6 busca exemplificar o sistema de inferência de Mamdani através do contexto de um pâncreas artificial.

Figura 6 – Inferência de Mamdani.



Fonte: (MAUSETH et al., 2013).

A princípio, revelam-se duas variáveis linguísticas: **BGL** e **BGL rate**. Ambas passam pelo processo de *fuzzyficação*, com valores 215 mg/dL e 0,9375 mg/dL/min, respectivamente. A primeira, revela um grau de pertinência de 0.5 com os conjuntos VH e H, nos quais as funções de pertinência são evidenciadas na imagem. A segunda variável revela um grau de pertinência de 0.25 com o conjunto Z e um grau de 0.75 com o conjunto P, nos quais as funções de pertinência são também evidenciadas pela imagem (MAUSETH et al., 2013).

A próxima fase trata-se da Avaliação das regras, na qual as entradas *fuzzificadas* são aplicadas aos antecedentes das regras fuzzy. Se uma dada regra difusa tiver múltiplos antecedentes, o operador difuso (*AND* ou *OR*) é usado para obter um único número que representa o resultado da avaliação do antecedente. Tipicamente, a operação de união é utilizada para o operador *OR* e a operação de interseção é utilizada para o operador *AND* (IANCU, 2012). Sendo assim, na imagem, os valores referentes aos graus de pertinência são mapeados na matriz de regras (MAUSETH et al., 2013).

Na fase de agregação, as funções de pertinência de todos os consequentes de regras previamente recortados ou dimensionados são combinados em um único conjunto difuso (IANCU, 2012). De acordo com o método direto de Mandani, lógica de operação mínima, o menor dos valores de BGL e taxa de BGL é selecionado. Então, a taxa de BGL Z (azul) e a BGL H (verde) emergem da matriz de regras (MAUSETH et al., 2013).

Entrando no estágio de *desfuzzificação*, os valores azul e verde são aplicados às suas respectivas coberturas difusas, produzindo dois polígonos que são combinados em um conjunto difuso, conforme mostrado no canto inferior direito. O centroide do conjunto é encontrado, o que produz uma dose nítida *desfuzzificada* de 0,76U (MAUSETH et al., 2013).

2.3 Aprendizado Profundo

O aprendizado profundo é um método de Inteligência Artificial (IA) que ensina computadores a processar dados de uma forma inspirada no cérebro humano. Os modelos de aprendizado profundo conseguem reconhecer padrões complexos em imagens, texto, sons e outros dados para produzir *insights* e previsões precisas. É possível usar métodos de aprendizado profundo para automatizar tarefas que normalmente exigem inteligência humana, como descrever imagens ou transcrever um arquivo de som em texto (AMAZON, 2025).

Através da introdução de diversas camadas de processamento, o aprendizado profundo adapta os modelos empregados à dificuldade do problema. Tipicamente, o aprendizado profundo requer modelos que capturem não-linearidades do problema, tornando a modelagem relativamente mais complexa (BOCHIE et al., 2020). O aprendizado profundo

engloba estruturas diversas que oferecem múltiplas camadas de processamento, porém as redes neurais concentram a maior parte dos esforços de pesquisa e dos algoritmos utilizados(BOCHIE et al., 2020).

2.3.1 Redes Neurais Artificiais (RNAs)

Uma rede neural é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse. Para alcançarem bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples denominadas “neurônios” ou “unidades de processamento”. O procedimento utilizado para realizar o processo de aprendizagem é chamado de algoritmo de aprendizagem, cuja função é modificar os pesos sinápticos da rede de uma maneira ordenada para alcançar um objetivo desejado (HAYKIN, 2001).

As redes neurais extraem seu poder computacional através de sua estrutura paralelamente distribuída e da sua habilidade de generalizar. Essas duas capacidades de processamento de informação tornam possível para as redes neurais resolver problemas complexos (HAYKIN, 2001).

2.3.2 Topologia das Redes Neurais

Em termos de topologia, para implementar uma rede neural deve-se determinar as seguintes variáveis (SPöRL; CASTRO; LUCHIARI, 2011):

- **Número de nós na camada de entrada:** correspondendo ao número de variáveis que serão usadas para alimentar a rede neural, geralmente as variáveis mais relevantes para o problema em estudo;
- **Número de camadas escondidas e número de neurônios nessas camadas:** normalmente, o número de camadas escondidas é bem reduzido pois possuem um melhor poder de generalização, reduzindo o *overfitting*. Todavia, se uma rede não tiver camadas escondidas o suficiente, pode levar a um *underfitting*, e
- **Número de neurônios na camada de saída:** estes parâmetros afetam o desempenho da RNA, devendo ser cuidadosamente escolhidos.

Ao desenvolver uma rede neural, outras decisões também devem ser tomadas, tais como:

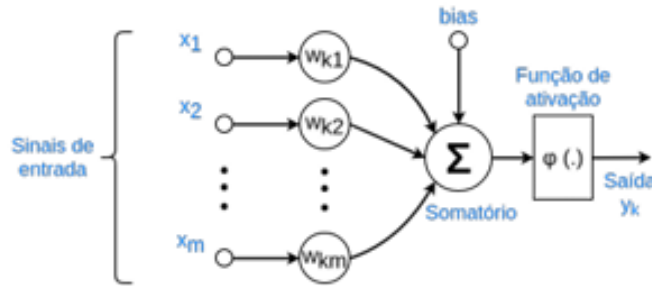
- Seleção da função de ativação dos neurônios da camada escondida e da camada de saída;
- Escolha do algoritmo de treinamento e seus respectivos parâmetros;

- Transformação dos dados ou método de padronização;
- Seleção do conjunto de treinamento e conjunto de teste;
- Critério de parada do treinamento, e
- Escolha de uma medida de desempenho da rede.

2.3.3 Modelos de um Neurônio

Um neurônio é uma unidade de processamento de informação que é fundamental para a operação de uma rede neural (HAYKIN, 2001). Estas unidades de processamento simples inicialmente foram definidas por McCulloch e Pitts em 1943, cujo aprendizado foi transformado em algoritmo por Charles Hebb em 1949 (KOPILER et al., 2019). A Figura 7 ilustra o modelo de um neurônio que forma a base para as redes neurais artificiais.

Figura 7 – Modelo Neural



Fonte: (BOCHIE et al., 2020).

Tratando a entrada como um vetor x , suas componentes $x_1, x_2, \dots, x_m$ correspondem às entradas do neurônio que são combinadas em uma soma ponderada, na qual esses valores são escalados utilizando seus respectivos pesos, representados por $w_{k1}, w_{k2}, \dots, w_{kn}$. O limiar de ativação inerente (*bias*) é introduzido como um grau de liberdade adicional para manipular a saída do neurônio, permitindo ao neurônio um melhor ajuste (BOCHIE et al., 2020). O resultado da soma entre o limiar de ativação e as entradas multiplicadas pelos pesos produz o potencial de ativação k . A saída, normalmente indicada pelo vetor y_k , corresponde ao potencial de ativação transformado pela função de ativação (BOCHIE et al., 2020), dada pela Equação 2.25:

$$y = \varphi(v); \quad (2.25)$$

2.3.4 Tipos de Função de Ativação

Segundo Haykin (2001), a função de ativação, representada por $\varphi(v)$, define a saída de um neurônio em termos do campo local induzido v . Pode-se identificar três tipos básicos

de função de ativação: Função Limiar, Função Linear por Partes e Função Sigmoidal.

2.3.4.1 Função Limiar

Para esse tipo de função, tem-se a Equação 2.26:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases} \quad (2.26)$$

Na literatura, essa função é conhecida como função de Heaviside. A definição $\varphi(v)$ descreve a propriedade tudo-ou-nada do modelo McCulloch-Pitts.

2.3.4.2 Função Linear por Partes

Para esse tipo de função, tem-se a Equação 2.27:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq \frac{1}{2} \\ v, & \text{se } -\frac{1}{2} \leq v \leq \frac{1}{2} \\ 0, & \text{se } v \leq -\frac{1}{2} \end{cases} \quad (2.27)$$

Esta forma de ativação pode ser vista como uma aproximação de um amplificador não linear. Nela, têm-se duas situações especiais:

1. Se a região linear de operação é mantida sem entrar em saturação, surge um combinador linear, e
2. Se fator de amplificação da região linear é grande, essa função se reduz à uma função limiar.

2.3.4.3 Função Sigmoidal

Para esse tipo de função, tem-se a Equação 2.28:

$$\varphi(v) = \frac{1}{1 + e^{-av}}; \quad (2.28)$$

Essa é a forma mais utilizada para a ativação de neurônios, equilibrando o processamento linear e não linear. O parâmetro a representa a inclinação da função sigmoide.

Todas as funções de ativação expostas seguem um intervalo de $[0, 1]$. Todavia, pode haver a necessidade de que ele seja $[-1, 1]$, assumindo uma forma anti-simétrica em relação à sua origem. Isso significa que a função de ativação é uma função ímpar do campo local induzido. Especificamente, pode ser definida pela Equação 2.29:

$$\varphi(v) = \begin{cases} 1, & \text{se } v > 0 \\ 0, & \text{se } v = 0 \\ -1, & \text{se } v < 0 \end{cases} \quad (2.29)$$

Essa é normalmente denominada função sinal. Para a forma correspondente de uma função sigmoide, pode-se utilizar a função tangente hiperbólica definida pela Equação 2.30:

$$\varphi(v) = \tanh(v); \quad (2.30)$$

O fato de permitir valores negativos nas funções de ativação traz benefícios analíticos. Com a rede normalizada, tal função contribui para um fluxo de gradiente mais rápida e estável em redes profundas. Além disso, em cenários onde a distribuição dos dados exige maior simetria, tais funções podem apresentar desempenho superior ao da ReLU (SINGH, 2024).

2.3.5 Aprendizagem e Treinamento de RNAs

A aprendizagem de uma rede neural é um processo onde os parâmetros livres são adaptados através de um processo de estimulação pelo ambiente em que a rede está inserida (HAYKIN, 2001). Este processo ocorre por meio do ajuste dos pesos (conexões entre neurônios) e dos limiares de ativação (*biases*), que são atualizados durante o treinamento. Os pesos influenciam a taxa de crescimento da função que o modelo tenta aprender; enquanto os *biases* deslocam a saída da função, permitindo maior flexibilidade (BOCHIE et al., 2020). Em síntese, tem-se a seguinte sequência de eventos para o aprendizado de uma rede:

1. A rede neural é estimulada por um ambiente;
2. A rede neural sofre modificações nos seus parâmetros livres como resultado desta estimulação, e
3. A rede neural responde de uma maneira nova ao ambiente, devido às modificações ocorridas na sua estrutura interna (HAYKIN, 2001).

Sendo assim, uma função linear do tipo $y = Tx + b$ pode ser modelada escolhendo adequadamente os pesos T e o *bias* b . Durante o treinamento, para cada amostra, o modelo calcula a saída com base nos parâmetros atuais e a compara com o valor alvo. A diferença entre o valor previsto e o valor real é quantificada por uma função de perda.

Essa perda, aplicada a várias amostras, é usada para compor uma função custo, que avalia o desempenho do modelo(BOCHIE et al., 2020).

A partir da função custo, calcula-se um vetor gradiente, que indica como cada parâmetro afeta a saída. Os pesos e *biases* são então ajustados no sentido oposto ao gradiente, utilizando uma taxa de aprendizado chamado *learning rate*, com o objetivo de minimizar o erro(BOCHIE et al., 2020).

Os neurônios são organizados em camadas: entrada, ocultas e saída. As redes neurais profundas destacam-se por conter múltiplas camadas ocultas. As camadas iniciais capturam características básicas; enquanto as mais profundas identificam padrões complexos, tornando o modelo capaz de realizar tarefas mais sofisticadas(BOCHIE et al., 2020).

2.3.6 Arquitetura de Redes Neurais

As disposições (arquitetura) de uma rede neural são variadas, conforme a necessidade da aplicação, estando diretamente ligadas ao algoritmo de aprendizagem utilizado para treinar a rede. Basicamente, os itens que compõem a estrutura de uma rede neural são sujeitos a modificações, sendo os seguintes(HAYKIN, 2001):

- Camadas intermediárias;
- Quantidade de neurônios;
- Função de transferência, e
- Algoritmo de aprendizado.

A maneira pela qual os neurônios de uma rede neural estão estruturados está intimamente ligada ao algoritmo de aprendizagem adotado. Pode-se, portanto, falar de algoritmos (ou regras) de aprendizagem como elementos estruturadores no projeto de redes neurais(MATSUNAGA, 2012). A seguir, são listadas as arquiteturas mais difundidas em redes neurais:

2.3.6.1 *FeedFoward Networks*, ou Redes Alimentadas Adiante

Nas Redes Neurais sem Realimentação (*Feedforward Neural Networks*), a informação é passada entre camadas em apenas um sentido a partir da entrada. O objetivo desse tipo de rede é aproximar uma função qualquer, definindo um mapeamento do tipo $y = f(x)$, onde a entrada x é conhecida e os parâmetros são ajustados para oferecer a melhor aproximação da função desejada (BOCHIE et al., 2020). Haykin (2001) distingue as redes alimentadas adiante em Redes de Camada Única e Redes de Múltiplas Camadas, diferenciando-se apenas pelo número de camadas, sendo:

- Os neurônios são arranjados em camadas, a camada inicial recebe os sinais de entrada e a camada final obtém as saídas. As camadas intermediárias são chamadas de camadas ocultas;
- Cada neurônio de uma camada é conectado com todos os neurônios da camada seguinte, e
- Não há conexões entre neurônios de uma mesma camada. A informação provém da camada de entrada ou neurônios fonte e, posteriormente, é transmitida para as camadas seguintes até a camada de saída. Este tipo de rede é também conhecido como rede acíclica, pois na representação de grafos não possui ciclos.

Uma rede neural sem realimentação é um aproximador universal. Isso quer dizer que essas redes têm a capacidade de representar qualquer função, dado que a rede neural seja suficientemente complexa. Uma rede neural sem realimentação clássica, com apenas uma camada oculta, é capaz de representar qualquer função mensurável, garantindo que a rede possua quantidade suficiente de neurônios na camada oculta. Entretanto, a literatura relata que para problemas complexos, o número de neurônios em apenas uma camada oculta é proibitivamente grande, o que torna a implementação de redes com essa arquitetura impraticável (BOCHIE et al., 2020).

2.3.6.2 Redes Convolucionais ou CNN

Uma Rede Neural Convolucional é uma variação das redes de Múltiplas Camadas, tendo sido inspirada no processo biológico de processamento de dados visuais. De maneira semelhante aos processos tradicionais de visão computacional, uma CNN é capaz de aplicar filtros em dados visuais, mantendo a relação de vizinhança entre os pixels da imagem ao longo do processamento da rede(VARGAS; PAES; VASCONCELOS, 2016).

Uma rede neural convolucional consiste em múltiplas partes com funções diferentes. Inicialmente, é comum aplicar sobre o dado de entrada camadas ditas de convolução. Uma camada de convolução é composta por diversos neurônios, cada um responsável por aplicar um filtro em um pedaço específico da imagem(VARGAS; PAES; VASCONCELOS, 2016).

Duas características das redes neurais convolucionais destacam-se, sendo: menor número de parâmetros e resistência a variações nos dados, como translação e distorção. A esparsidade de conexões entre os neurônios de camadas adjacentes e o compartilhamento de parâmetros são fundamentais para tornar mais eficiente a retirada de características locais dos dados. Dessa forma, torna-se mais difícil que apenas pedaços da rede neural se especialize em uma tarefa; enquanto outros pedaços são subutilizados (BOCHIE et al., 2020).

2.3.6.3 Redes Recorrentes

A estrutura de uma rede neural recorrente tem por base uma rede *feedforward* com algumas modificações, notadamente a introdução de realimentação, o que amplia sua potencialidade de modelagem de dados temporais ou espaciais(GOMES, 2005).

Através das conexões recorrentes, a rede tem acesso a estados anteriores, isto é, valores inseridos e computados em iterações passadas. Isso permite que RNNs mapeiem qualquer sequência em outra sequência dado um número suficiente de unidades ocultas, conferindo um resultado equivalente à aproximação universal visto nas redes neurais sem realimentação (BOCHIE et al., 2020)

O compartilhamento de estados requer a introdução de novos pesos à rede para possibilitar que a saída da rede seja calculada em função das entradas, do estado da rede e das saídas anteriores. Além da diferença na computação dos resultados, também é importante mencionar que, em contraste com as entradas de tamanho fixo nas redes neurais de *feedforward* e CNN, a entrada de uma RNN é feita de forma gradual e não possui limitação quanto ao tamanho(BOCHIE et al., 2020).

Uma limitação comum das redes neurais recorrentes é a dificuldade no aprendizado de dependências de longo prazo. Sendo assim, as RNNs tradicionais têm dificuldade de computar a influência de estados muito distantes da iteração atual. Durante o aprendizado em redes que usam sequências longas, os gradientes tendem a desaparecer ou explodir, fenômenos chamados de *vanishing gradient* e *exploding gradient* respectivamente, inviabilizando a capacidade da rede de aprender as dependências de longo prazo(BOCHIE et al., 2020).

2.3.6.4 Redes Neurais Autoassociativas ou *Autoencoder*

O objetivo da Rede Neural Autoassociativa é produzir na saída uma representação que seja a mais próxima possível do dado inserido na entrada da rede. Para isso, ela utiliza um ajuste de pesos não supervisionado, uma vez que a qualidade dos pesos é verificada a partir da análise da reconstrução do dado comparado ao valor inserido anteriormente (BOCHIE et al., 2020).

O objetivo do aprendizado em uma rede autoassociativa com estrutura codificador-decodificador é ajustar os pesos para gerar uma representação compacta e relevante dos dados, que permita sua reconstrução. A simples cópia dos dados não é útil, pois não contribui para a extração de informações significativas (BOCHIE et al., 2020).

2.3.7 *Long Short-Term Memory (LSTM)*

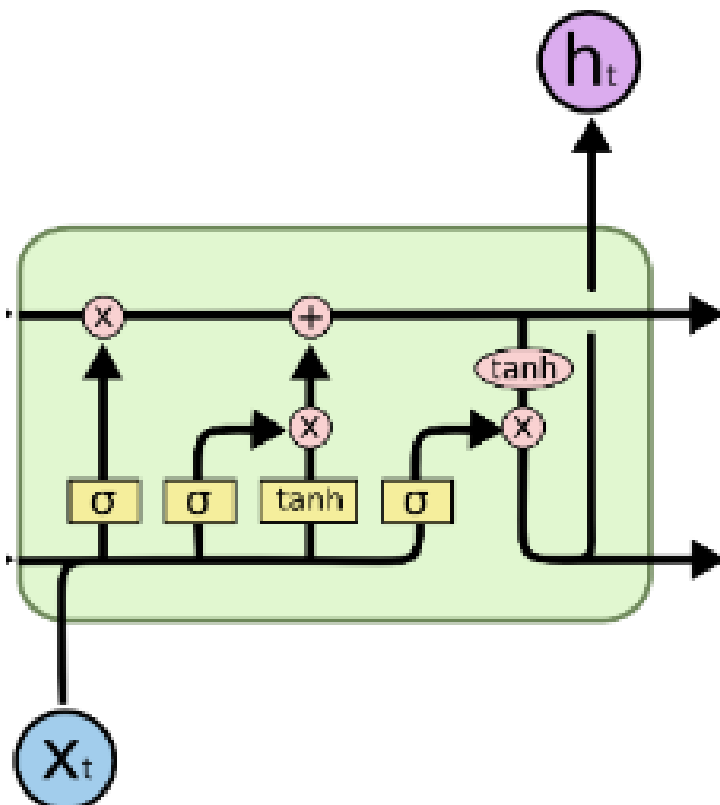
Long Short-Term Memory, ou *LSTM*, é uma *RNN* que surgiu para resolver os problemas de gradiente encontrados criado por Hochreiter e Schmidhuber (1997). Eles

aprimoraram a capacidade de memorização da célula recorrente padrão, introduzindo os chamados portões ou *gates* na célula (YU et al., 2019). Os mesmos são estruturas que permitem a passagem de informações, através de uma camada de rede neural sigmoide e uma operação de multiplicação pontual(OLAH, 2015). Desde esse trabalho pioneiro, as *LSTMs* foram modificadas e popularizadas por muitos pesquisadores.(YU et al., 2019).

A camada oculta da *LSTM* é composta por blocos de memória, que são sub-redes autoconectadas contendo múltiplas células internas. Por meio de portas multiplicativas, a célula é capaz de armazenar e acessar informações por um longo período de tempo (GRAVES et al., 2009).

Em outras palavras, a *LSTM* carrega dados de várias etapas por todas as etapas, e cada etapa da célula é capaz de incluir e remover informações desses dados enquanto processa entradas sequenciais(DRUMOND et al., 2018). A Figura 8 evidencia a arquitetura de um bloco *LSTM*, cada linha carrega um vetor inteiro, da saída de um nó às entradas de outros. Os círculos rosa representam operações pontuais, como adição de vetores; enquanto as caixas amarelas representam camadas de redes neurais aprendidas. Linhas que se fundem denotam concatenação; Enquanto uma bifurcação de linha denota que seu conteúdo está sendo copiado e as cópias indo para locais diferentes(OLAH, 2015).

Figura 8 – Arquitetura de um bloco *LSTM*.



Fonte: (OLAH, 2015).

Segundo Olah (2015), o funcionamento de um bloco *LSTM* pode ser descrito no seguinte passo a passo:

1. **Decisão do que esquecer:** O primeiro passo é decidir quais informações do estado anterior da célula devem ser descartadas. Essa decisão é feita por uma camada sigmoide chamada *porta de esquecimento* (*forget gate*). Essa porta analisa o vetor de saída anterior h_{t-1} e a entrada atual x_t , produzindo um vetor de valores entre 0 e 1 para cada número no estado da célula C_{t-1} . Valores próximos de 0 indicam que a informação deve ser esquecida; enquanto valores próximos de 1 indicam que deve ser mantida. Tem-se a Equação 2.31:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f); \quad (2.31)$$

2. **Decisão do que armazenar:** Em seguida, decide-se quais novas informações serão armazenadas no estado da célula. Esse processo envolve duas etapas:

- Uma *porta de entrada* (*input gate*), que é uma camada sigmoide responsável por indicar quais valores serão atualizados, e
- Uma camada de ativação tangente hiperbólica que gera um vetor de candidatos a novos valores $\tilde{C}_t$, que podem ser adicionados ao estado da célula.

As equações associadas a esse passo são representadas por 2.32 e 2.33:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \quad (2.32)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C); \quad (2.33)$$

3. **Atualização do estado da célula:** Agora, é feita a atualização do estado da célula anterior C_{t-1} para o novo estado C_t . Isso é feito combinando as informações antigas, ponderadas pela porta de esquecimento, com as novas informações, ponderadas pela porta de entrada, conforme a Equação 2.34

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t; \quad (2.34)$$

4. **Geração da saída:** Por fim, decide-se o que será enviado como saída da célula. Primeiro, uma porta sigmoide chamada *porta de saída* (*output gate*) decide quais partes do estado da célula serão incluídas na saída. Em seguida, o novo estado da célula passa por uma função $\tanh$, e é multiplicado pelos valores da porta de saída para gerar a saída final h_t , conforme as Equações 2.35 e 2.36

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o); \quad (2.35)$$

$$h_t = o_t * \tanh(C_t); \quad (2.36)$$

O segredo das *LSTMs* é o estado de cada célula, a linha horizontal que passa pelo topo do diagrama, percorrendo a cadeia e passando por pequenas interações lineares. Desta forma, a informação pode fluir por ela sem alterações (OLAH, 2015).

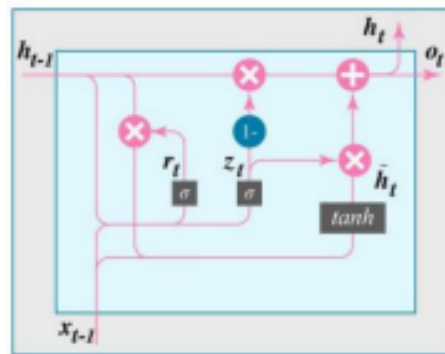
Além dessa estrutura básica, as *LSTMs* podem contar com variações, como por exemplo as chamadas *peephole connections*. Nesse modelo, as portas têm acesso direto ao estado interno da célula (C_t), permitindo um controle mais preciso sobre o fluxo de informação (OLAH, 2015).

2.3.8 Gated Recurrent Unit (GRU)

As GRUs foram introduzidas em 2014 como uma alternativa para diminuir a complexidade das unidades LSTM (NOSOUHIAN; NOSOUHIAN; KHOSHOUEI, 2021). A GRU combina a porta de entrada e a porta de esquecimento da LSTM em uma única porta de atualização, resultando em um *design* mais simplificado. Ao contrário da LSTM, a GRU não inclui um estado de célula separado (SHIRI et al., 2023).

Uma unidade GRU consiste em três componentes principais: uma porta de atualização, uma porta de reinicialização e o conteúdo atual da memória. Essas portas permitem que a GRU atualize e utilize seletivamente informações de passos de tempo anteriores, permitindo capturar dependências de longo prazo em sequências (SHIRI et al., 2023). A Figura 9 apresenta a estrutura de uma unidade GRU.

Figura 9 – Arquitetura de um bloco *GRU*.



Fonte: (SHIRI et al., 2023).

Segundo (SHIRI et al., 2023), o cálculo ocorre da seguinte forma:

1. **Porta de atualização:** determina quanto das informações passadas deve ser retido e combinado com a entrada atual em um passo de tempo específico. Ela é calculada com base na concatenação do estado oculto anterior h_{t-1} e da entrada atual x_t , seguida por uma transformação linear e uma função de ativação sigmoide, segundo

a Equação 2.37:

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z); \quad (2.37)$$

2. **Porta de reinicialização:** decide quanto da informação passada deve ser esquecida. Ela é computada de forma semelhante, a partir da concatenação do estado oculto anterior e da entrada atual, segundo a Equação 2.38:

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r); \quad (2.38)$$

3. **Conteúdo atual da memória:** é calculado com base na porta de reinicialização e na concatenação do estado oculto transformado e da entrada atual. O resultado passa por uma função de ativação tangente hiperbólica para produzir a ativação candidata, segundo a Equação 2.39:

$$\tilde{h}_t = \tanh(W_h[r_t \cdot h_{t-1}, x_t]); \quad (2.39)$$

4. **Estado final da memória:** h_t é determinado pela combinação do estado oculto anterior e da ativação candidata, regulada pela porta de atualização, segundo a Equação 2.40:

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t; \quad (2.40)$$

5. **Porta de saída (opcional):** controla o fluxo de informações do estado oculto para a saída. Ela é computada a partir do estado atual da memória h_t , segundo a Equação 2.41:

$$o_t = \sigma(W_o h_t + b_o); \quad (2.41)$$

onde W_o e b_o representam, respectivamente, a matriz de pesos e o vetor de polarização da camada de saída.

A GRU oferece uma alternativa mais simples à LSTM, com menos operações tensoriais e, portanto, treinamento mais rápido. No entanto, a escolha entre GRU e LSTM depende do caso de uso específico e do problema em questão, já que ambas apresentam vantagens e desvantagens (SHIRI et al., 2023).

2.3.9 Métricas

Para a análise de redes neurais, pode-se utilizar as seguintes métricas:

1. **Erro Médio Absoluto (EMA)**

É uma métrica que avalia a precisão do modelo, calculando a média das diferenças absolutas entre os valores observados e os valores preditos pelo modelo. Deve-se buscar sempre um valor de EMA menor, pois isso indica que os valores preditos

estão próximos dos ótimos (WANG; LU, 2018). Sendo representada pela Equação 2.42:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|; \quad (2.42)$$

onde:

- n é o número total de amostras;
- y_i é o valor real da amostra i , e
- $\hat{y}_i$ é o valor previsto da amostra i .

2. Raiz do Erro Quadrático Médio (RMSE)

Também chamado de *Root Mean Square Error*, é uma métrica utilizada para medir a precisão do modelo. Calcula-se a raiz quadrada da média das diferenças quadradas entre os valores observados e os valores preditos. Assim como o EMA, deve-se buscar minimizá-lo (WANG; LU, 2018). Sendo representada pela Equação 2.43:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}; \quad (2.43)$$

onde:

- n é o número total de amostras;
- y_i é o valor real da amostra i , e
- $\hat{y}_i$ é o valor previsto da amostra i .

3. Coeficiente de Determinação (R^2)

É o quadrado do coeficiente de correlação amostral. Retorna a porcentagem da variabilidade das variáveis que se deseja prever. Quando seu valor é próximo de um, indica que a nuvem de pontos em um diagrama de dispersão está próxima da reta de regressão; quanto mais próximo de zero, mais difícil é vislumbrar uma estrutura linear. É utilizado para medir a adequação do modelo de regressão ao conjunto de dados (MARTINS, 2018). Sendo representada pela Equação 2.44:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}; \quad (2.44)$$

onde:

- n é o número total de amostras;
- y_i é o valor real da amostra i ;
- $\hat{y}_i$ é o valor previsto da amostra i , e
- $\bar{y}$ é a média dos valores reais.

2.4 Domain Driven Design

Segundo Fowler ([FOWLER, 2020](#)), *Domain-Driven Design*, ou apenas DDD, é uma abordagem para desenvolvimento de *software* que centraliza o desenvolvimento na programação de um modelo de domínio que tem um rico entendimento dos processos e regras de um domínio. A abordagem é particularmente adequada para domínios complexos, onde muita lógica frequentemente confusa precisa ser organizada.

Princípios

Segundo Evans ([EVANS, 2004](#)), o DDD baseia-se nos seguintes princípios:

1. Linguagem Ubíqua

Um projeto enfrenta problemas com a fragmentação de linguagem. Desta maneira, a mesma linguagem deve ser usada para diagramas, escrita e especialmente fala para se referir a algo durante o projeto. Com uma linguagem ubíqua, o modelo não é apenas um artefato de *design*, mas se torna parte integrante de tudo o que os desenvolvedores e especialistas de domínio fazem juntos. A linguagem carrega conhecimento em uma forma dinâmica;

2. *Bounded Contexts*

Múltiplos contextos podem trazer *bugs* para o código. Desta maneira, deve-se definir explicitamente o contexto dentro do qual um modelo se aplica, mantendo assim o contexto estritamente consistente dentro dos limites do mesmo, e

3. *Context Maps*

Todos os *Bounded Contexts* de um projeto devem ser mapeados de forma clara, utilizando a Linguagem Ubíqua. Além disso, é importante incluir os modelos implícitos presentes em subsistemas que não adotam a abordagem orientada a objetos. Esse mapeamento auxilia na compreensão das interações entre os diferentes contextos e na mitigação de potenciais conflitos dentro do sistema.

2.4.1 Novos Conceitos

Evans ([EVANS, 2004](#)) introduz em seu livro conceitos que são definidos por [Fowler \(2007\)](#) como:

1. *Entities*

Objetos que têm uma identidade distinta que atravessa o tempo e representações diferentes. Conhecidos como "*Reference Objects*";

2. *Value Objects*

Objetos que importam apenas como a combinação de seus atributos. Dois objetos de valor com os mesmos valores para todos os seus atributos são considerados iguais, e

3. *Service Objects*

Uma operação autônoma dentro do contexto do seu domínio. Um objeto de serviço coleta um ou mais serviços em um objeto. Normalmente, haverá apenas uma instância de cada tipo de objeto de serviço dentro do seu contexto de execução.

Com estes novos conceitos, [Evans \(2004\)](#) preencheu uma lacuna importante no pensamento sobre objetos que não eram consideradas nem nas linguagens de programação, nem nas notações diagramáticas ([FOWLER, 2020](#)).

2.5 Arquitetura Hexagonal

A arquitetura Hexagonal surge como uma abordagem para o desenvolvimento de aplicações agnósticas em relação às tecnologias. O seu objetivo é permitir que um aplicativo seja controlado da mesma forma por usuários, programas, testes automatizados ou *scripts* em lote, sendo desenvolvido e testado isoladamente de seus dispositivos de tempo de execução e banco de dados ([COCKBURN, 2009](#)).

De acordo com [Cockburn \(2009\)](#), nessa arquitetura, a lógica de aplicação fica totalmente encapsulada dentro do núcleo e, para que a informação passe por ele, deve ser submetida a duas estruturas principais situadas nas fronteiras da arquitetura, sendo as Portas e os Adaptadores.

Portas

São protocolos e interfaces que definem como a aplicação funciona e quais dados ela necessita, possibilitando a conexão com o mundo externo. As portas podem ser divididas em duas classes:

- **Portas principais**, ou adaptadores condutores, que conduzem o usuário a uma ação, e
- **Portas secundárias**, ou adaptadores prolongados, que levam o usuário a uma ação.

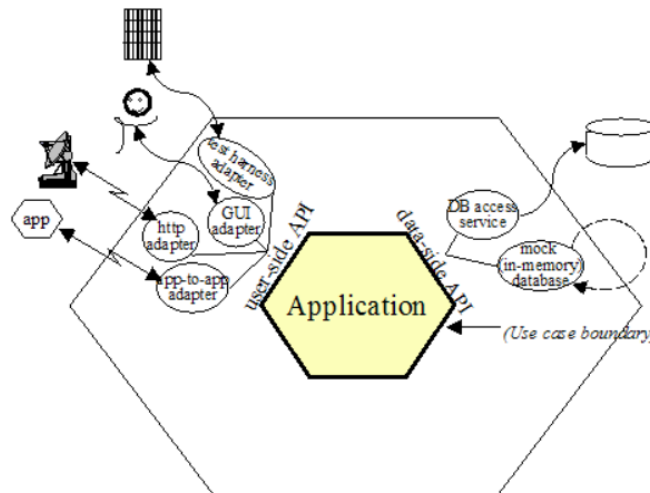
Nos diagramas, as portas principais aparecem à esquerda, e as portas secundárias à direita, respectivamente.

Adaptadores

Os adaptadores estão presentes em cada porta e conectam cada elemento da aplicação com o mundo externo.

Tais especificações são ilustradas na Figura 10:

Figura 10 – Arquitetura Hexagonal



Fonte: (COCKBURN, 2009).

Segundo Cockburn (2009), criador desta arquitetura, nela estão presentes alguns padrões de projeto, como:

- **Adapter:** Padrão de projeto responsável por converter a interface de uma classe para a interface esperada pelo cliente. Este padrão permite que duas classes incompatíveis trabalhem em conjunto (GAMMA et al., 1995);
- **Injeção de dependência:** A ideia básica do padrão é ter um objeto montador separado que preenche um campo em uma classe com uma implementação apropriada para a interface (FOWLER, 2004); e
- **Mock Objects:** Objetos especiais que imitam objetos reais para teste (FOWLER, 2007).

2.6 Resumo do Capítulo

Este capítulo apresenta os principais conceitos utilizados no desenvolvimento e implementação deste trabalho. Inicialmente, são abordados os fundamentos de sistemas de controle, fornecendo uma visão geral sobre o equilíbrio de variáveis em um sistema. Em seguida, é introduzida a lógica difusa, que, por meio de variáveis linguísticas, busca produzir uma saída discreta. Posteriormente, são discutidas as redes neurais, que atuarão como

complemento à lógica difusa. Por fim, são explorados os conceitos de DDD e arquitetura hexagonal, visando aprimorar a qualidade do código e a organização do sistema.

3 Suporte Tecnológico

Este capítulo aborda as principais ferramentas, organizadas com base em seu tipo de apoio. As [Ferramentas de Apoio à Pesquisa](#) são aquelas que contribuíram para a realização do processo de levantamento bibliográfico e orientação deste trabalho. As [Ferramentas de Apoio Desenvolvimento](#) são aquelas que auxiliaram na implementação das provas de conceito inerentes à proposta deste trabalho. Na sequência, compreende-se as [Ferramentas de Apoio à Elaboração da Monografia](#), as quais auxiliaram na escrita deste documento. Por conseguinte, pode-se encontrar as [Ferramentas de Apoio Complementar](#), que correspondem aos meios de comunicação entre orientando e orientadores. Por fim, compreende-se o [Resumo do Capítulo](#).

3.1 Ferramentas de Apoio à Pesquisa

Esta seção elenca as principais ferramentas utilizadas no processo de levantamento bibliográfico desta pesquisa. Essas ferramentas também foram utilizadas para prover controle de prazos e como caderno de anotações. Seguem as principais:

- ***Obsidian***: editor *Markdown* projetado para organizar informações de maneira estruturada e interconectada. Permite controle sobre os dados, armazenando localmente e possibilitando integração com *Dropbox*, *Git* e criptografia de disco ([OBSIDIAN, 2025](#)). Foi utilizado para elaboração de resumos, anotações gerais e cronogramas.
- ***Google Docs***: ferramenta de edição e criação de documentos armazenados online([DOCS, 2025](#)). Utilizado para anotações gerais.
- ***Adobe Acrobat Reader***: ferramenta essencial para criação, edição, revisão e assinatura eletrônica de arquivos PDF([READER, 2025](#)). Utilizada para leitura de materiais teóricos e visualização do produto final documental da monografia.
- ***Trello***: Ferramenta para organizar projetos através de quadros([TRELLO, 2025](#)). Utilizada para o desenvolvimento do quadro *Kanban*.

3.2 Ferramentas de Apoio ao Desenvolvimento

Esta seção evidencia as ferramentas de desenvolvimento das Provas de Conceito, as quais em conjuntos compõem o software proposto neste trabalho. Seguem as principais:

- **Simglucose:** Simulador de pacientes com Diabetes tipo 1. Este simulador é uma implementação em *Python* do Simulador *UVa/Padova* (versão 2008), aprovado pela FDA, para fins exclusivamente de pesquisa. O simulador inclui 30 pacientes virtuais, 10 adolescentes, 10 adultos e 10 crianças(XIE, 2018).

O Modelo simula os níveis de glicose e insulina no sangue e como essas substâncias entram, circulam e saem do corpo, levando em conta: a glicose produzida pelo corpo, pela comida ingerida e parte da insulina eliminada pelos rins, além disso, evidencia a velocidade com que a mesma entra na corrente sanguínea. Também evidencia como a insulina é absorvida e destruída depois de ser aplicada no corpo(MAN et al., 2014).

O simulador divide a glicose em um modelo de dois compartimentos: o primeiro refere-se ao sangue e aos tecidos em equilíbrio rápido; O segundo compartimento refere-se aos tecidos em equilíbrio lento. Agora, o simulador descreve a insulina também com dois compartimentos, um representado pelo fígado e outra pelo plasma(MAN et al., 2014).

Todavia, quando a insulina é aplicada de um meio externo, ou seja, subcutânea, este processo é dividido em três compartimentos; (i) a produção natural de glicose do corpo é realizada linearmente na qual cresce dependendo da quantidade de glicose no sangue e de um sinal atrasado da insulina; (ii) a glicose dos alimentos é absorvida pelo corpo após passar pelo estômago e pelo intestino; (iii) durante a refeição, o corpo utiliza a glicose para o cérebro e os glóbulos vermelhos, sem insulina. Já para os tecidos, dependendo da quantidade de glicose, com ajuda da insulina(MAN et al., 2014).

O simulador também incorpora erros de medição associados ao CGM, simulando leituras imprecisas com base nos valores reais de glicose. Esse recurso permite uma representação mais realista do comportamento do sensor. Todos os testes dos modelos propostos foram realizados utilizando esse simulador.

- **Python:** Linguagem de programação interpretada, multiparadigma, amplamente reconhecida pela sua versatilidade, poder e simplicidade de aprendizado (PYTHON, 2025). Neste trabalho, *Python* foi a principal linguagem utilizada para o desenvolvimento das provas de conceito, devido à sua ampla adoção na comunidade acadêmica e industrial, especialmente nas áreas de ciência de dados e aprendizado de máquina. A escolha de *Python* também foi motivada pela familiaridade do autor com a linguagem, o que facilita o processo de desenvolvimento e reduz a curva de aprendizado. Além disso, o simulador utilizado no projeto já estava implementado em *Python*, o que tornava a integração entre os módulos mais eficiente. Para o desenvolvimento do sistema de controle baseado em lógica difusa, foi empregada a biblioteca *Skfuzzy*;

enquanto a biblioteca *TensorFlow* foi utilizada para o treinamento e validação da rede neural proposta.

- **Visual Studio Code:** Um ambiente de desenvolvimento integrado (IDE) compatível com *Linux*, *macOS* e *Windows*, e oferece suporte a uma ampla gama de linguagens de programação (MICROSOFT, 2021). A escolha dessa ferramenta se deu pela familiaridade do autor com o *software*, o que proporciona um ambiente de desenvolvimento eficiente e intuitivo. O mesmo foi utilizado para o desenvolvimento e testes das provas de conceito, com o auxílio de extensões específicas para *Python*, que otimizaram o processo de execução e depuração dos códigos.
- **Google Colab:** É um serviço baseado no *Jupyter Notebook*, que oferece acesso gratuito a recursos avançados como *GPUs* e *TPUs*, sendo amplamente utilizado em projetos de aprendizado de máquina e ciência de dados (COLAB, 2025). Considerando as limitações de *hardware* do autor, essa plataforma foi escolhida devido ao seu acesso gratuito a máquinas de alto desempenho, permitindo o treinamento e validação da rede neural proposta de forma eficiente. Além disso, o *Google Colab* facilita a integração com as principais bibliotecas utilizadas no trabalho, como *TensorFlow*, e oferece facilidades no processamento paralelo, tornando-o ideal para experimentos de aprendizado de máquina.
- **Google Sheets:** É uma ferramenta para criação de planilhas online que possibilita a colaboração em tempo real (SHEETS, 2025). Neste trabalho, foi utilizada para armazenar e organizar os dados extraídos do simulador, os quais foram posteriormente processados pela rede neural. O uso do *Google Sheets* facilitou a manipulação e análise preliminar dos dados, proporcionando uma solução prática para o controle e visualização das informações coletadas ao longo do projeto.
- **Git/GitHub:** *Git* é um sistema de controle de versão distribuído e gratuito (GIT, 2025), enquanto *GitHub* é uma plataforma online de hospedagem de código, que facilita o versionamento e a colaboração no desenvolvimento de projetos (GITHUB, 2025). Durante o desenvolvimento das provas de conceito, ambas as ferramentas foram utilizadas para versionar o código, garantindo que todas as alterações fossem armazenadas de maneira segura e acessível. O uso do *Git* e do *GitHub* permitiu um gerenciamento eficiente das diferentes versões do código, facilitando a colaboração e o rastreamento das modificações. O código das provas de conceito pode ser acessado neste link¹.

¹ Código disponível em: <https://github.com/kaua-pt/TCC_ArtificialPancreas>, acesso em 06 de novembro de 2025.

3.3 Ferramentas de Apoio à Elaboração da Monografia

Nesta seção, podem ser encontradas as ferramentas utilizadas para a escrita desta monografia, com destaque para:

- **Lucidchart**: ferramenta para criação de diagramas diversos([LUCIDCHART, 2025](#)). Utilizada para desenhar a arquitetura das provas de conceito;
- **LaTeX**: sistema de composição tipográfica de alta qualidade, voltado para documentação técnica e científica([LATEX, 2025](#)). Utilizado na redação do presente documento, e
- **OverLeaf**: editor *LaTeX* online com colaboração em tempo real e controle de versão([OVERLEAF, 2025](#)). Utilizado na escrita desta monografia.

3.4 Ferramentas de Apoio Complementar

Nesta seção, encontram-se as ferramentas auxiliares para o desenvolvimento deste trabalho. Essas correspondem aos meios de comunicação entre os orientadores e o orientando.

- **Slack**: plataforma de comunicação que conecta pessoas, processos, dados e *IA* em uma interface conversacional([SLACK, 2025](#)). Usado como principal meio de comunicação;
- **Teams**: ferramenta do *Microsoft 365* para colaboração em equipe([TEAMS, 2025](#)). Utilizada para reuniões periódicas, e
- **Telegram**: aplicativo de mensagens baseado em nuvem, com foco em segurança e agilidade([TELEGRAM, 2025](#)). Usado para avisos pontuais durante o projeto.

3.5 Resumo do Capítulo

Neste capítulo, foram abordadas as principais ferramentas tecnológicas utilizadas para o desenvolvimento deste trabalho, além das suas devidas provas de conceito. A seguir, a Tabela 1 demonstra como estão listadas as ferramentas em questão, bem como as versões utilizadas.

Tabela 1 – Resumo das Ferramentas Utilizadas

Nome	Classe da Ferramenta	Versão
<i>Obsidian</i>	Apoio à pesquisa	2025
<i>Google Docs</i>	Apoio à pesquisa	2025
<i>Adobe Acrobat Reader</i>	Apoio à pesquisa	2025
<i>Trello</i>	Apoio à pesquisa	2024
<i>Python</i>	Apoio ao desenvolvimento	3.13
<i>Simglucose</i>	Apoio ao desenvolvimento	v0.2.1
<i>Visual Studio Code</i>	Apoio ao desenvolvimento	2021
<i>Google Colab</i>	Apoio ao desenvolvimento	2025
<i>Google Sheets</i>	Apoio ao desenvolvimento	2025
<i>Git/GitHub</i>	Apoio ao desenvolvimento	2025
<i>Lucidchart</i>	Apoio à elaboração da monografia	2025
<i>LaTeX</i>	Apoio à elaboração da monografia	2
<i>Overleaf</i>	Apoio à elaboração da monografia	2025
<i>Slack</i>	Apoio complementar	25.04.10.0
<i>Teams</i>	Apoio complementar	1416/1.0.0.2025062402
<i>Telegram</i>	Apoio complementar	11.9.0

Fonte: Autor.

4 Metodologia

Este capítulo descreve os métodos que norteiam as atividades desenvolvidas neste estudo. Apresenta-se a [Classificação da Pesquisa](#), evidenciando abordagem, natureza, objetivos e procedimentos da pesquisa. Na sequência, é descrito o método investigativo, compreendendo o [Levantamento Bibliográfico](#). Detalha-se ainda sobre o uso de [Provas de Conceito](#), visando facilitar o desenvolvimento desse trabalho, bem como sua apresentação para outros interessados, conforme consta em Método Orientado a Provas de Conceito. Têm-se o [Método de Desenvolvimento](#) orientando-se por práticas ágeis, bem como o [Método de Análise de Resultados](#), sendo esse centrado em provas de conceito e Pesquisa-Ação. Adiante, consta o [Cronograma das Atividades](#) desenvolvidas ao longo da pesquisa. Por fim, é apresentado o [Resumo do Capítulo](#).

4.1 Classificação da Pesquisa

De acordo com [Gerhardt e Silveira \(2009\)](#), a pesquisa científica é o resultado de um inquérito ou exame minucioso, realizado com o objetivo de resolver um problema, recorrendo a procedimentos científicos. Assim, seguindo as definições dos mesmos, este trabalho é classificado nos seguintes aspectos: **Abordagem, Natureza, Objetivos e Procedimentos**. A Tabela 2 resume essa classificação, e na sequência são apresentadas suas respectivas justificativas.

Tabela 2 – Classificação da Pesquisa.

Aspecto	Classificação
Abordagem	Quantitativa
Natureza	Aplicada
Objetivos	Exploratória
Procedimentos	Levantamento Bibliográfico, Provas de Conceito e Pesquisa-Ação

Fonte: Autor.

4.1.1 Abordagem de Pesquisa

Quanto à abordagem da pesquisa, esta pode ser classificada como quantitativa, conforme apontam [Gerhardt e Silveira \(2009\)](#). Os resultados obtidos são, assim, considerados um retrato real da população-alvo. A abordagem quantitativa destaca-se pela objetividade e pelo uso da linguagem matemática para descrever as causas de um fenômeno e as relações entre variáveis ([FONSECA, 2002](#)).

Neste trabalho, adotou-se exclusivamente a abordagem quantitativa. A utilização de lógica difusa e redes neurais reflete uma orientação baseada em dados mensuráveis, alinhada aos princípios dessa abordagem.

4.1.2 Objetivo de Pesquisa

Esta pesquisa pode ser classificada como exploratória, pois tem como objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito (GERHARDT; SILVEIRA, 2009). Desta maneira, o trabalho busca informar os interessados sobre como construir um pâncreas artificial, *in silico*, que possa se adaptar às necessidades de gerenciamento de glicose do paciente.

4.1.3 Natureza de Pesquisa

Trata-se de uma pesquisa aplicada, pois tem como objetivo gerar conhecimento para uma aplicação prática, afim de solucionar um problema (GERHARDT; SILVEIRA, 2009). Tal problema traduz-se no desenvolvimento de um pâncreas artificial voltado ao gerenciamento da glicose em pacientes com diabetes tipo 1. Para isso, empregam-se técnicas de Inteligência Artificial (IA) e lógica difusa.

4.1.4 Procedimentos de Pesquisa

Os procedimentos de pesquisa utilizados neste trabalho são orientados ao levantamento bibliográfico, à Pesquisa-Ação e à Prova de conceito. O primeiro ajudou na investigação de conceitos e princípios, permitindo acordar uma base teórica sólida para esse trabalho. Já o segundo consiste em uma abordagem prática para a validação de conceitos teóricos, com o intuito de demonstrar a viabilidade de uma solução proposta (KOTTAPALLE et al., 2021a). As provas de conceito foram utilizadas para validar todas as observações propostas pelos artigos que embasam esse trabalho.

4.2 Levantamento Bibliográfico

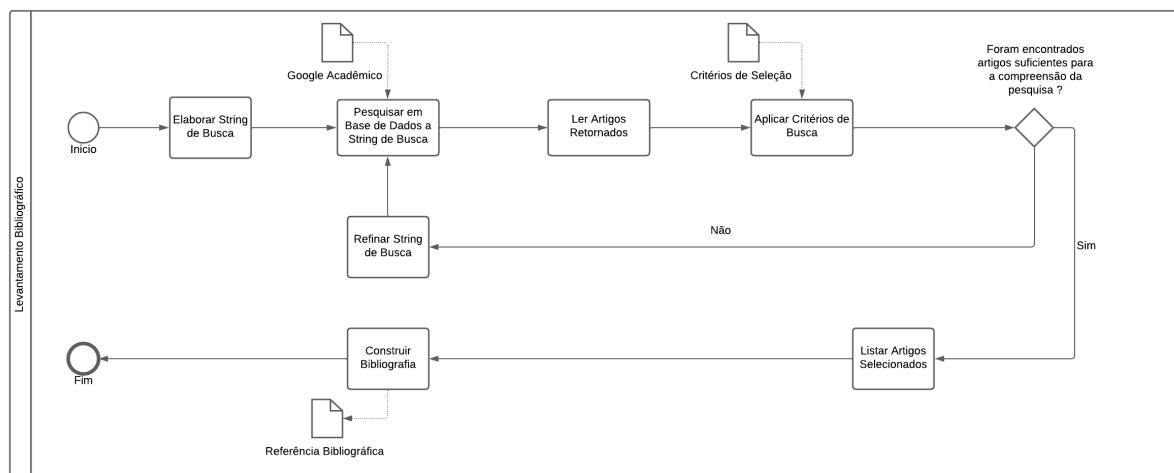
O levantamento bibliográfico visa proporcionar ao pesquisador uma base teórica sólida sobre o tema abordado (GIL, 2002). Assim, após a definição do tema, realizou-se uma pesquisa bibliográfica para embasar o desenvolvimento do estudo. A Figura 11 ilustra o diagrama, na notação BPMN¹, referente ao levantamento bibliográfico realizado.

Basicamente, têm-se as atividades: Elaborar *String* de Busca, sendo essa formada pelas principais palavras-chave orientadas aos tópicos de interesse desse trabalho; Pesqui-

¹ BPMN (business process model and notation) é uma maneira padronizada de demonstrar processos completos que ocorrem em um negócio. Disponível em <<https://www.ibm.com/br-pt/think/topics/bpmn>>. Último acesso: 07/05/2025

sar em Base de Dados a *String* de Busca, ou seja, usar o Google Acadêmico como base de consulta; Ler Artigos Retornados, sendo lidos os mais relevantes (i.e. os mais citados e mais recentes, limitando-se às primeiras duas páginas de retorno); Aplicar Critérios de Busca, sendo vários, conforme esclarecido na seção 4.2.2; Refinar *String* de Busca, caso ainda não tenham sido encontrados artigos suficientes; Listar Artigos Seleccionados, caso já se tenha um conjunto de artigos razoável para compreensão da pesquisa, e Construir Bibliografia, resultando nas Referências Bibliográficas que fundamentam essa pesquisa (seção 4.2.3), além de outros complementares apresentados ao final desta monografia.

Figura 11 – BPMN Levantamento Bibliográfico.



Fonte: Autor.

4.2.1 *Strings* de Busca

As *strings* de busca foram refinadas ao longo do tempo para obter artigos mais relevantes. A Tabela 3 apresenta as principais combinações utilizadas.

4.2.2 Critérios de Seleção

A seleção dos artigos foi baseada na análise do resumo e das palavras-chave, adotando-se os critérios a seguir. Lembrando que foram considerados os artigos mais citados e mais recentes, limitando-se às primeiras duas páginas de retorno.

- Relação com diabetes tipo 1;
- Identificação de vulnerabilidades do paciente diabético tipo 1;
- Aplicação de lógica difusa;

Tabela 3 – *Strings* de Busca Utilizadas.

<i>String</i> de Busca	Base de Dados	Resultados
Artificial AND pancreas	Google Scholar	653.000
Fuzzy AND artificial AND pancreas	Google Scholar	16.000
Insulin AND sensibility AND diabetes AND type AND 1	Google Scholar	29.300
Fuzzy AND logic	Google Scholar	2.200.000
Controller AND fuzzy AND logic	Google Scholar	658.000
Artificial AND pancreas AND fuzzy AND logic	Google Scholar	7.980
Error AND insulin AND artificial AND pancreas	Google Scholar	39.900
Artificial AND pancreas AND type AND 1 AND diabetes	Google Scholar	275.000
Type AND 1 AND diabetes	Google Scholar	4.000.000
Artificial AND pancreas AND personal	Google Scholar	91.200
Artificial AND pancreas AND algorithm	Google Scholar	74.700
Pancreas AND artificial	Google Scholar	912.000
Pancreas AND artificial AND lógica AND difusa	Google Scholar	680
Diabetic AND routine	Google Scholar	1.620.000
Diabetes AND type AND 1 AND alimentation	Google Scholar	33.000
Lógica AND fuzzy AND conceitos	Google Scholar	21.400
Sistemas AND fuzzy	Google Scholar	80.600
Redes AND neurais AND artificiais	Google Scholar	59.300
Lstm	Google Scholar	1.240.000
Redes AND neurais	Google Scholar	99.300
Funções AND ativação AND redes AND neurais	Google Scholar	25.000
Engenharia AND controle AND moderno	Google Scholar	211.000
Sistemas AND controle	Google Scholar	3.790.000
Sistemas AND controle AND malha AND fechada	Google Scholar	39.200
redes AND neurais AND topologia	Google Scholar	13.600
gru	Google Scholar	788.000
gru AND lstm	Google Scholar	236.000

Fonte: Autor.

- Aplicação de redes neurais;
- Aplicação de conceitos de sistemas de controle;
- Linguagem matemática clara e compreensível;
- Enfoque em Inteligência Artificial;
- Disponibilidade gratuita, e
- Relevância na área.

4.2.3 Artigos Selecionados

A partir desse processo, os seguintes artigos foram considerados fundamentais para o estudo:

- *Fuzzy Sets*([ZADEH, 1965](#));
- *Proposed Clinical Application for Tuning Fuzzy Logic Controller of Artificial Pancreas Utilizing a Personalization Factor*([MAUSETH et al., 2010](#));
- *Use of a “Fuzzy Logic” Controller in a Closed-Loop Artificial Pancreas* ([MAUSETH et al., 2013](#));
- *Stress Testing of an Artificial Pancreas System With Pizza and Exercise Leads to Improvements in the System’s Fuzzy Logic Controller* ([MAUSETH et al., 2015](#));
- *Recent advances in the precision control strategy of artificial pancreas*([MING et al., 2024](#));
- *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*([CHEN; PHAM, 2000](#)), e
- *Redes Neurais: Princípios e Prática*([HAYKIN, 2001](#)).

4.3 Método Orientado a Provas de Conceito

No desenvolvimento de software, o termo Prova de conceito pode ser uma ferramenta essencial para demonstrar capacidades e adequação do software conforme a necessidade dos clientes. As provas de conceito são projetadas para determinar se um conceito de software pode ser criado no mundo real, além de ajudar a determinar qual tecnologia pode ser utilizada para desenvolvimento e comprovar se os usuários alvo realmente utilizarão o software([KOTTAPALLE et al., 2021b](#)).

Neste trabalho, as provas de conceito são em conjunto com um simulador de diabetes tipo 1, e orientando-se pelas informações contidas nos artigos base.

Para desenvolvimento das provas de conceito, foi utilizado o método sugerido por ([KOTTAPALLE et al., 2021b](#)), consistindo em 4 estágios:

1. Problema

Momento no qual deve-se definir a imagem-alvo mensurável para que o projeto possa ser avaliado quando pronto;

2. Solução Ótima

Etapas que se preocupa com as oportunidades, explorando-as de forma estratégica para maximizar suas vantagens. As oportunidades podem surgir do conhecimento de projetos anteriores, bem como podem ser identificadas no campo ou em áreas previamente inexploradas. Neste estágio, é importante escolher uma solução abstrata

que criará o maior benefício, sem especificações técnicas, mas descrevendo a solução em um nível geral;

3. Potencial

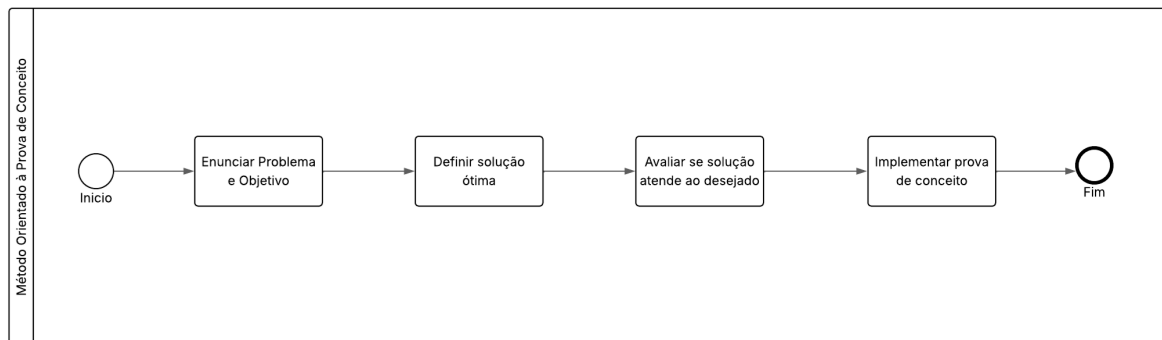
Etapa relevante para o andamento do projeto. Deve-se avaliar se a solução alcançada atende à definição do projeto e à meta. Deve-se evitar custos desnecessários que surgem ao desenvolver a prova de conceito. Portanto, é importante determinar se o projeto tem potencial ou não; e

4. *PoC Designing*

Etapa voltada para a produção propriamente dita da prova de conceito. Nela, ocorre o levantamento de insumos, definição da arquitetura, implementação e avaliação da prova de conceito.

A Figura 12 ilustra o diagrama, na notação BPMN, referente ao método orientado às provas de conceito.

Figura 12 – BPMN Método Orientado às Provas de Conceito.



Fonte: Autor.

4.4 Método de Desenvolvimento

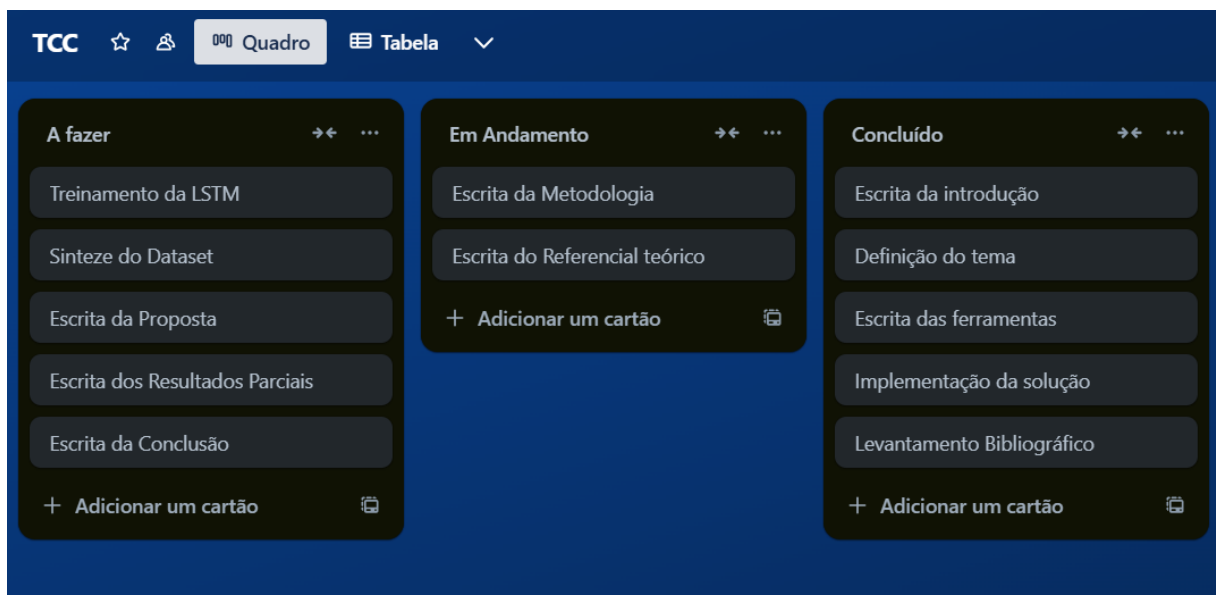
Para as atividades de desenvolvimento, como é o caso do desenvolvimento das provas de conceito, foi utilizado o método *Kanban* (BOEG, 2010). O *Kanban*, ou mais precisamente o "sistema *Kanban* para desenvolvimento de software", representa uma implementação mais direta dos princípios de Desenvolvimento *Lean* de Produtos do que a forma comumente realizada pelos métodos ágeis tradicionais. Com foco consistente no fluxo e no contexto, o *Kanban* oferece uma abordagem menos prescritiva comparada ao *Agile*, e tem se tornado uma extensão popular dos métodos ágeis tradicionais como *Scrum* e *XP* (BOEG, 2010).

Existem diversas abordagens para o *Kanban*. Como um método de gestão de mudanças, dá-se ênfase aos seguintes princípios:

- Visualizar o trabalho em andamento;
- Visualizar cada passo em sua cadeia de valor, do conceito geral até um incremento de software que se possa lançar;
- Limitar o Trabalho em Progresso (WIP – *Work in Progress*), restringindo o total de trabalho permitido para cada estágio;
- Tornar explícitas as políticas sendo seguidas;
- Medir e gerenciar o fluxo, para poder tomar decisões bem embasadas, além de visualizar as consequências dessas decisões, e
- Identificar oportunidades de melhorias, criando uma cultura *Kaizen*, na qual a melhoria contínua é responsabilidade de todos (BOEG, 2010). Entende-se por "todos", nesse projeto solo, como sendo o autor, os orientadores, os autores das fontes de referência desse trabalho, dentre outros envolvidos e interessados no mesmo.

Desta maneira, a Figura 13 evidencia a utilização da ferramenta trello para o desenvolvimento tanto da primeira prova de conceito, quanto da devida monografia.

Figura 13 – Quadro Kanban.



Fonte: Autor.

O quadro Kanban foi importante para a organização do projeto, conferindo ainda clareza sobre o andamento do mesmo.

4.5 Método de Análise de Resultados

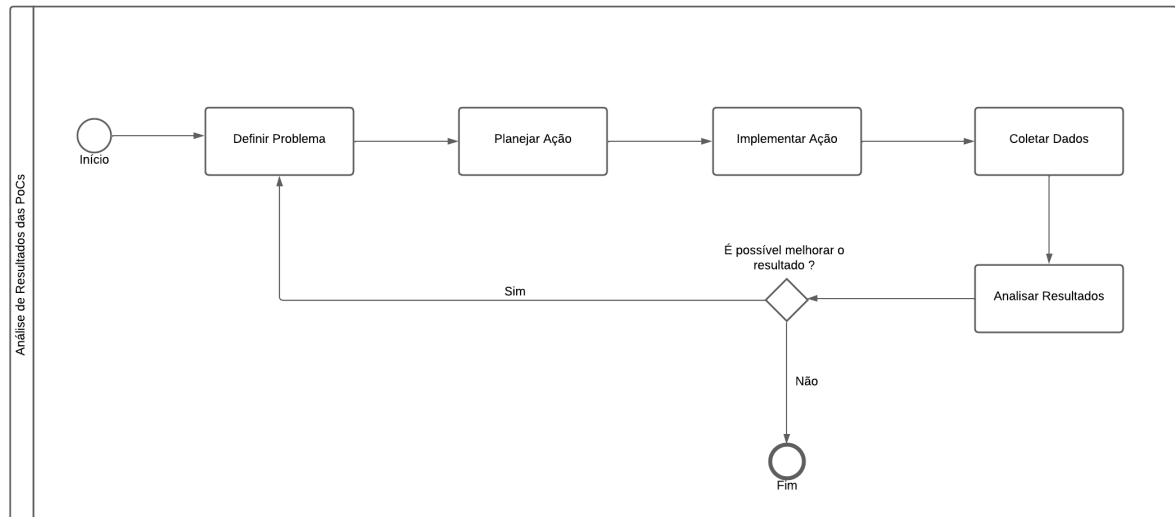
Ao se desenvolver provas de conceito, torna-se necessário realizar avaliações quantitativas, visando aferir se a implementação está de acordo com as informações adquiridas via literatura e materiais de base. Segundo [Kottapalle et al. \(2021b\)](#), uma prova de conceito tem como objetivo testar uma solução em seu ambiente de aplicação, além de coletar dados e analisar os resultados obtidos. Nesse sentido, buscou-se pelo uso de um método complementar. O método escolhido foi pesquisa-ação, no qual procura-se unir pesquisa à ação prática, rompendo lacunas entre ambos ([ENGEL, 2000](#)).

Segundo [Engel \(2000\)](#), a pesquisa ação é um método cíclico composto das seguintes fases:

1. **Definição do problema:** Consiste na identificação de um aspecto intrigante, passível de aprimoramento ou com potencial inovador. Após essa identificação, cada ponto levantado deve ser submetido a uma análise preliminar, a fim de avaliar sua relevância e viabilidade;
2. **Pesquisa preliminar:** Consiste na revisão bibliográfica e no levantamento de necessidades relacionadas ao problema identificado;
3. **Formulação de hipóteses:** Consiste na elaboração de hipóteses que poderão ser testadas ao longo do desenvolvimento da pesquisa. Deve-se levar em consideração os dados coletados na fase anterior;
4. **Desenvolvimento de um plano de ação:** Consiste na definição de estratégias e na elaboração de um plano sistemático para solucionar o problema ou atingir os objetivos propostos;
5. **Implementação do plano de ação:** Consiste na execução do plano previamente elaborado, seguindo as estratégias especificadas;
6. **Coleta de dados:** Consiste na obtenção de informações e evidências que sirvam de subsídio para a análise do problema. Os dados coletados são submetidos a métricas específicas, permitindo uma compreensão mais precisa da situação;
7. **Avaliação do plano de execução:** Consiste na realização de uma análise crítica, seguida da interpretação dos resultados, visando tirar conclusões fundamentadas, bem como verificar se o plano de ação produziu os efeitos esperados, e
8. **Comunicação dos resultados:** Consiste na divulgação dos achados e das experiências obtidas ao longo do processo, tornando os resultados acessíveis à comunidade interessada.

A Figura 14 ilustra o diagrama BPMN referente ao processo de análise de resultados realizado efetivamente, com foco nas ações Definir Problema, Planejar Ação, Implementar Ação, Coletar Dados e Analisar Resultados.

Figura 14 – BPMN Método de Análise de Resultados.



Fonte: Autor.

4.6 Cronogramas de Atividades

Esta seção descreve os cronogramas de atividades correspondentes à realização da primeira e segunda etapas do TCC. Cada atividade está acompanhada de seu respectivo período de realização, status e resultado.

4.6.1 Atividades e Subprocessos da Primeira Etapa do TCC

A seguir, serão descritas as atividades correspondentes à primeira etapa desse trabalho. Tais atividades correspondem ao período entre Março a Julho de 2025.

1. Definir do Tema

A atividade consistiu na identificação de áreas de interesse do autor com relevância na área de software. Como resultado, foi proposto o tema da monografia;

- **Período:** Março;
- **Status:** Realizada; e
- **Resultado:** Tema definido: *"In Silico: Desenvolvimento de um Pâncreas Artificial com Lógica Difusa e Aprendizado Profundo"*.

2. Levantamento Bibliográfico

Subprocesso destinado à construção da base teórica do trabalho. A partir dela, foram desenvolvidas a seção de [Levantamento Bibliográfico](#), bem como a [Justificativa](#) e os [Questionamentos](#) da Pesquisa;

- **Período:** Março a Abril;
- **Status:** Realizada; e
- **Resultado:** [Levantamento Bibliográfico](#), [Justificativa](#) e [Questionamentos](#).

3. Formular Proposta Inicial

Responsável por definir os objetivos a serem alcançados ao longo do trabalho, tanto o [Objetivo Geral](#); Quanto os [Objetivos Específicos](#);

- **Período:** Março a Abril;
- **Status:** Realizada; e
- **Resultado:** [Objetivo Geral](#) e [Objetivos Específicos](#) formulados.

4. Definir Cronograma de Atividades

Elaboração do [Cronograma](#) de execução das atividades relacionadas à primeira etapa do trabalho;

- **Período:** Março a Abril;
- **Status:** Realizada; e
- **Resultado:** [Cronograma](#) estabelecido.

5. Desenvolvimento Inicial das Provas de Conceito

Subprocesso, orietando aos Métodos Provas de Conceito e de Desenvolvimento, para implementar a versão inicial do projeto, com o intuito de aprofundar o entendimento das problemáticas envolvidas e possíveis soluções relacionadas ao tema;

- **Período:** Março a Abril;
- **Status:** Realizada; e
- **Resultado:** [Provas de Conceito](#).

6. Elaborar Introdução

Compreende a redação do [Capítulo 1: Introdução](#), incluindo [Contextualização](#), [Justificativa](#), [Questionamento](#) e [Objetivos](#);

- **Período:** Abril;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 1](#) finalizado.

7. Elaborar Referencial Teórico

Redação do [Capítulo 2: Referencial Teórico](#), abordando os principais conceitos relacionados ao tema, como [Sistemas de Controle](#), [Lógica Difusa](#), [Redes Neurais](#), [Domain-Driven Design](#) e [Arquitetura Hexagonal](#);

- **Período:** Abril;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 2](#) finalizado.

8. Elaborar Suporte Tecnológico

Produção do [Capítulo 3: Suporte Tecnológico](#), no qual são descritas as ferramentas e tecnologias utilizadas no processo de desenvolvimento e documentação do projeto;

- **Período:** Abril;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 3](#) finalizado.

9. Elaborar Metodologia

Redação do [Capítulo 4: Metodologia](#), que apresenta os métodos utilizados para a condução da pesquisa e para o desenvolvimento do projeto;

- **Período:** Abril a Maio;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 4](#) finalizado.

10. Elaborar Proposta

Produção do [Capítulo 5: Proposta \(Monografia da Primeira Etapa do TCC\)](#), o qual aprofunda a Contextualização do problema e apresenta como os conceitos discutidos no [Referencial Teórico](#) serão aplicados para responder ao [Questionamento](#) e atingir os [Objetivos](#) propostos;

- **Período:** Abril a Maio;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 5: Proposta \(Monografia da Primeira Etapa do TCC\)](#) finalizado.

11. Elaborar Resultados Parciais

Redação do [Capítulo 6: Resultados Parciais \(Monografia da Primeira Etapa do TCC\)](#), constando os avanços e resultados alcançados ao longo da primeira etapa do TCC;

- **Período:** Maio a Junho;

- **Status:** Realizada; e
- **Resultado:** Capítulo 6: Resultados Parciais (Monografia da Primeira Etapa do TCC) finalizado.

12. Revisar Artefatos

Revisão e correção de todos os capítulos previamente desenvolvidos, em preparação para a apresentação à banca avaliadora;

- **Período:** Maio a Julho;
- **Status:** Realizada; e
- **Resultado:** Monografia (Primeira Etapa do TCC) revisada e formatada.

13. Apresentar Primeira Etapa do TCC

Realização da apresentação formal do trabalho à banca avaliadora, conforme os critérios estabelecidos pela instituição;

- **Período:** Julho;
- **Status:** Realizada; e
- **Resultado:** Apresentação do trabalho para a banca.

A Figura 15 ilustra o diagrama, em notação BPMN, referente ao Cronograma da primeira etapa do TCC realizado

4.6.2 Atividades e Subprocessos da Segunda Etapa do TCC

A seguir, serão descritas as atividades correspondentes à segunda etapa desse trabalho. Tais atividades correspondem ao período entre Agosto a Dezembro de 2025.

1. Aplicar Correções da Primeira Etapa do TCC

Implementação das correções sugeridas pelos membros da banca avaliadora na primeira etapa do trabalho;

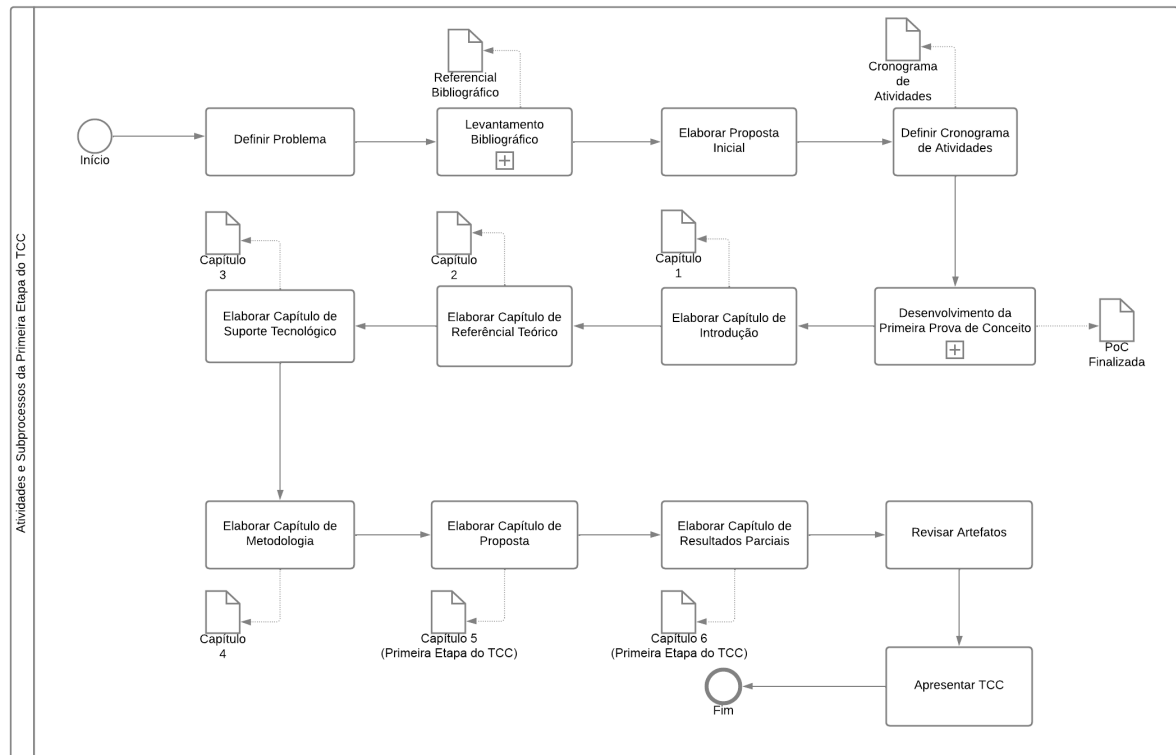
- **Período:** Agosto;
- **Status:** Realizada; e
- **Resultado:** Correções aplicadas.

2. Aprimorar Provas de Conceito

Aprimoramento das provas de conceito com base nas sugestões da banca, além da continuidade do desenvolvimento com novos recursos;

- **Período:** Agosto e Setembro;

Figura 15 – BPMN de Atividades e Subprocessos da Primeira Etapa do TCC.



Fonte: Autor.

- **Status:** Realizada; e
- **Resultado:** [Provas de Conceito](#).

3. Análise de Resultados

Avaliação dos resultados obtidos nas provas de conceito por meio de métricas quantitativas;

- **Período:** Outubro;
- **Status:** Realizada; e
- **Resultado:** Resultados analisados e provas de conceito ajustadas.

4. Documentar Resultados

Redação do capítulo referente aos resultados finais obtidos com o desenvolvimento das provas de conceito;

- **Período:** Outubro;
- **Status:** Realizada; e
- **Resultado:** Capítulo de [Capítulo 6: Análise de Resultados](#) concluído.

5. Documentar Conclusão

Elaboração do capítulo de conclusão, destacando o estado final do projeto, as atividades realizadas e considerações finais;

- **Período:** Novembro;
- **Status:** Realizada; e
- **Resultado:** [Capítulo 7: Conclusão](#) escrito.

6. Revisar Artefatos

Revisão e ajustes de todos os capítulos previamente desenvolvidos, em preparação para a apresentação final à banca avaliadora;

- **Período:** Novembro a Dezembro;
- **Status:** Realizada; e
- **Resultado:** Artefatos Revisados - Presente Monografia.

7. Apresentar Segunda Etapa do TCC

Apresentação formal da monografia à banca avaliadora, conforme os critérios estabelecidos pela instituição;

- **Período:** Dezembro;
- **Status:** Não iniciada; e
- **Resultado:** Apresentação realizada.

8. Aplicar Correções Finais

Aplicar correções finais conforme solicitado pela banca.

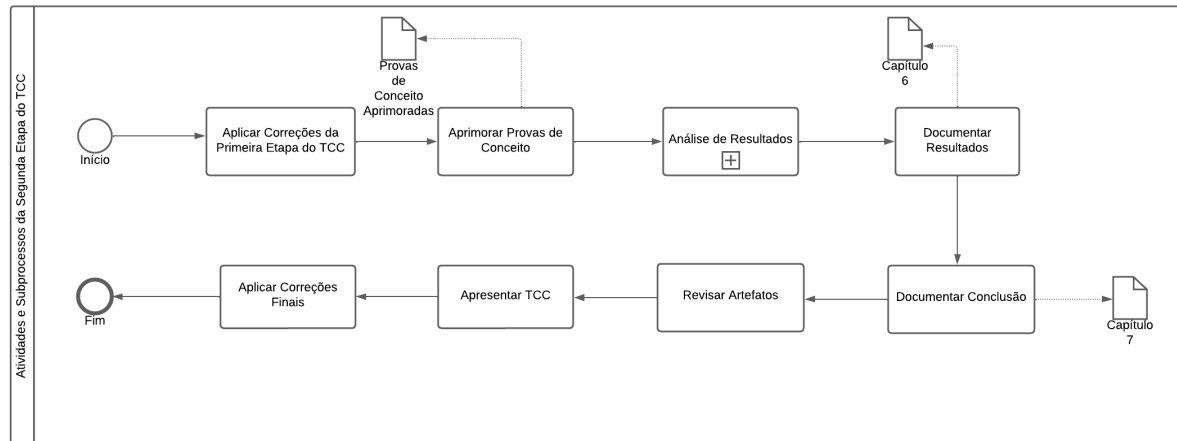
- **Período:** Dezembro;
- **Status:** Não iniciada; e
- **Resultado:** Monografia finalizada.

A Figura 16 ilustra do diagrama, em notação BPMN, referente ao Cronograma da segunda etapa do TCC realizado.

4.7 Resumo do Capítulo

Neste capítulo, foram abordados os principais aspectos metodológicos que fundamentaram o desenvolvimento deste trabalho. Inicialmente, apresentou-se a classificação da pesquisa, a qual, quanto à abordagem, caracteriza-se como quantitativa, com natureza

Figura 16 – BPMN de Atividades e Subprocessos da Segunda Etapa do TCC.



Fonte: Autor.

aplicada. No que diz respeito aos objetivos, a pesquisa é classificada como exploratória, tendo sido conduzida com base em procedimentos fundamentados em levantamento bibliográfico, provas de conceito e pesquisa-ação.

Em seguida, descreveu-se o processo de levantamento bibliográfico, no qual foram detalhadas as *strings* de busca; os critérios de seleção de documentos, e os principais artigos considerados essenciais para a fundamentação teórica do estudo. Posteriormente, foi discutido o método orientado à prova de conceito, explicando-se sua aplicação prática, e a forma como tem sido estruturado e implementado neste trabalho. Há ainda um descritivo sobre o método de pesquisa-ação, que complementa a análise de resultados. Por fim, apresentou-se o cronograma de atividades, contemplando a descrição de cada tarefa, o período de execução, o status e os respectivos resultados alcançados.

5 Pâncreas Artificial

Neste capítulo, será apresentado produto desenvolvido, o Pâncreas Artificial. Primeiramente, a seção de [Contextualização](#) introduzirá o tema com uma abordagem voltada para a Engenharia de Software. Em seguida, serão descritas as [Provas de Conceito \(PoCs\)](#) desenvolvidas, abordando elementos como Lógica Difusa e Inteligência Artificial, bem como a forma como esses temas foram aplicados nas POCs. Por fim, será apresentado o [Resumo do Capítulo](#).

5.1 Contextualização

Este trabalho surgiu da proximidade do autor com o tema, motivado pela convivência com familiares portadores de diabetes. Ao observar a rotina desafiadora e os cuidados intensivos exigidos para a estabilização da doença, emergiu a motivação para investigar soluções que minimizem a carga de informações necessárias por parte do paciente e de sua família para cumprir este objetivo, buscando alternativas que ofereçam maior autonomia e qualidade de vida, chegando assim na ideia do desenvolvimento de um pâncreas artificial.

Desta maneira, [Ming et al. \(2024\)](#) apresentam ideias de como seria a arquitetura de um pâncreas artificial, e como poderia ser construído um controlador para um sistema de malha fechada, no qual, se destacam os controladores de Inteligência Artificial que podem fornecer cálculos personalizados de dosagens de insulina sob demanda. Com isso, [Mauseth et al. \(2010\)](#) propuseram, um controlador de lógica difusa que depende apenas de parâmetros de gerenciamento de glicose, que foram pré-determinados por um clínico especialista em diabetes e codificados para uso no controlador.

Para o desenvolvimento do trabalho em andamento, os testes foram feitos unicamente no simulador *Simglucose*, que já foi discorrido no [Capítulo 3](#).

O controlador de lógica difusa proposto tem como foco a administração de insulina em um circuito fechado, com o objetivo de reduzir a ocorrência e/ou a duração de episódios de hipoglicemia e hiperglicemia. Para a avaliação do Pâncreas Artificial, serão utilizadas as métricas discorridas na seção de [Sistemas de Controle](#).

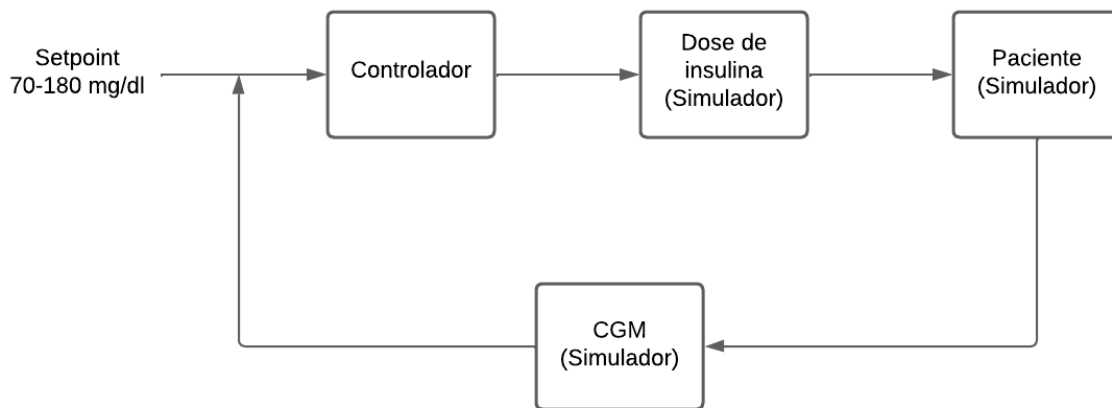
5.2 Provas de Conceito

Como evidenciado no [Capítulo 4](#), na seção [Método Orientado a Prova de Conceito](#), as Provas de Conceito são ferramentas essenciais para demonstrar as capacidades de um

software conforme a necessidade.

A princípio, vale ressaltar que um Pâncreas Artificial consiste em um [Sistema de Malha Fechada](#), que se utiliza de um *Continuous Glucose Monitoring* (CGM) e uma bomba de insulina para mimetizar a função biológica do pâncreas em indivíduos com diabetes tipo 1. Desta maneira, o Sistema de malha Fechada da PoC ocorre conforme a Figura 17:

Figura 17 – Sistema de Malha Fechada das POCs.



Fonte: Autor.

O subsistema de *Continuous Glucose Monitoring*(CGM) fornece, a cada instante, os valores de glicemia estimados pelo simulador. Esses valores são encaminhados ao controlador, que os compara com um valor de referência (*setpoint*). A partir dessa comparação, o controlador realiza calculos utilizando-se de técnicas de Lógica Difusa e Inteligência Artificial para determinar a dose de insulina a ser administrada ao paciente no tempo t . O simulador, então, calcula o efeito dessa insulina no organismo do paciente virtual e atualiza o estado metabólico do mesmo. Em seguida, uma nova medição de glicose é realizada, reiniciando o ciclo de controle.

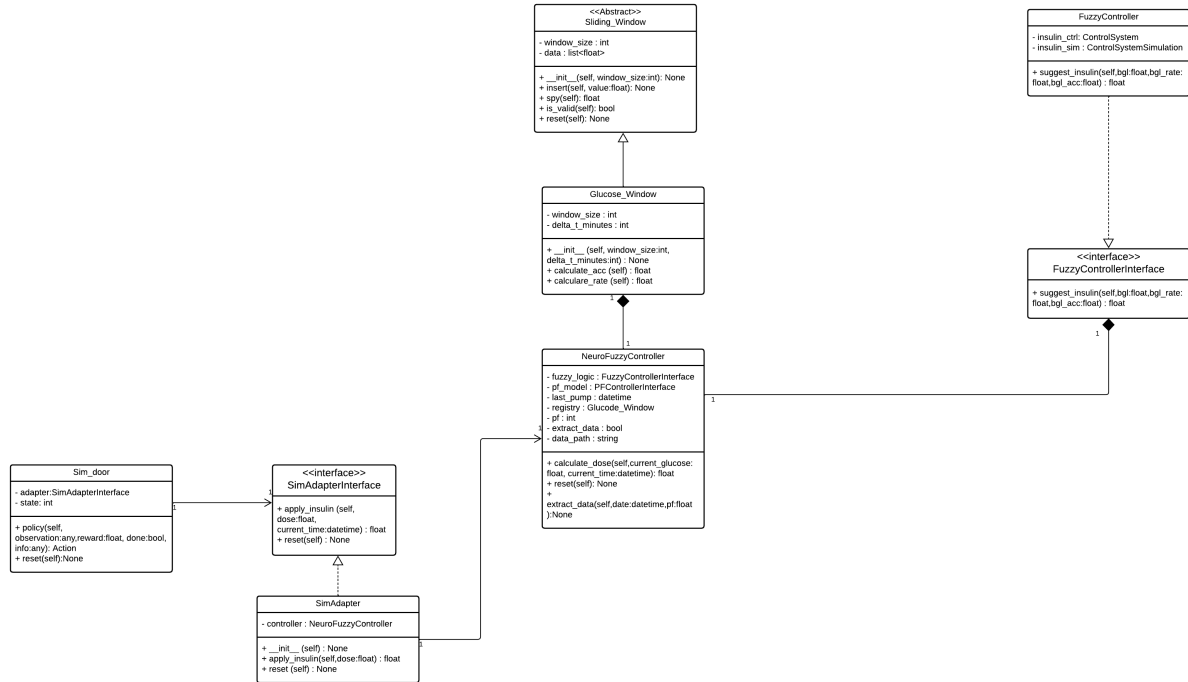
Foram realizadas duas provas de conceito (POCs) para o desenvolvimento do controlador: a primeira utilizando apenas a lógica difusa, e a segunda combinando a lógica difusa com técnicas de Inteligência Artificial, com a finalidade de compará-las e, adicionalmente, cumprir com os objetivos elencados no [Capítulo 1](#), na seção de [Objetivos](#).

Cada prova de conceito segue a [Arquitetura de Portas e Adaptadores](#). A escolha dessa arquitetura visa possibilitar futuras substituições de simuladores sem que haja a necessidade de alterar aspectos internos do código. Essa flexibilidade é garantida pelo uso do padrão de projeto *Adapter* discorrido no [Capítulo 2](#).

O diagrama de classes correspondente à primeira prova de conceito é apresentado

na Figura 18; enquanto o diagrama da segunda prova de conceito é apresentado na Figura 19.

Figura 18 – Diagrama de Classes da Primeira POC.



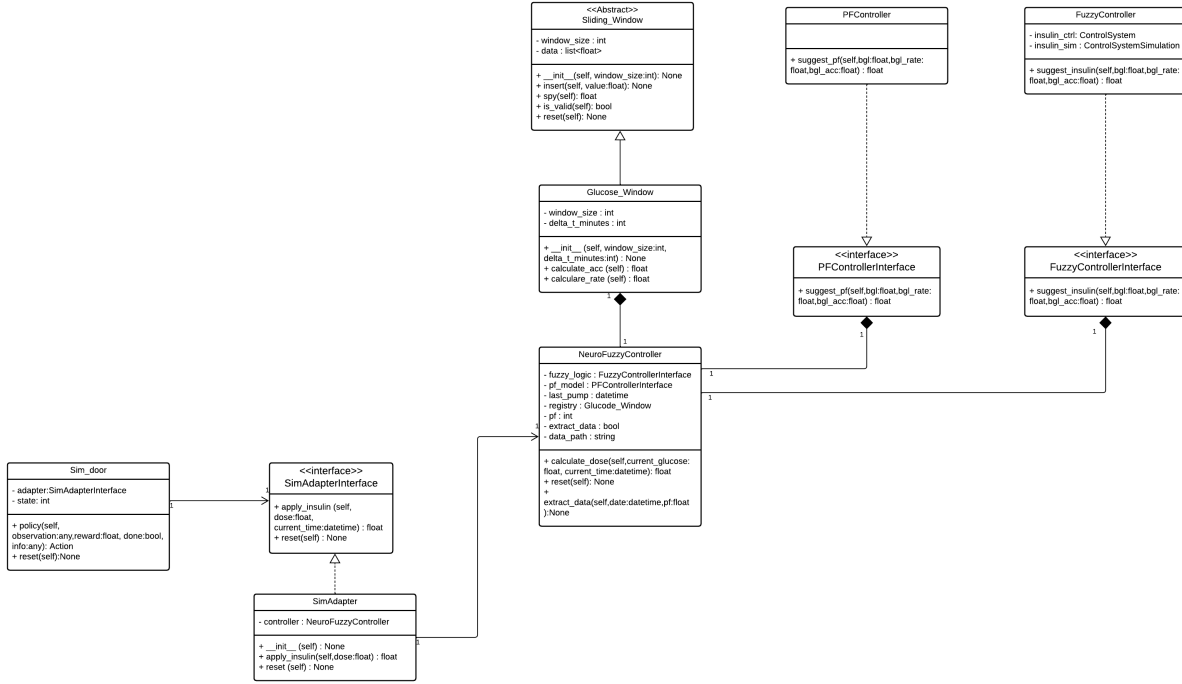
Fonte: Autor.

Ambas as modelagens apresentam, entre seus elementos, um adaptador primário, representado inicialmente pela classe *Sim_door*, responsável por interagir com o simulador. Essa classe está associada à interface *SimAdapterInterface*, que tem como função padronizar a entrada de valores provenientes do simulador. A classe *SimAdapter* implementa essa interface, possibilitando a comunicação com a classe principal do controlador, a *NeuroFuzzyController*.

Caso seja necessário substituir o simulador utilizado, basta criar uma nova classe concreta que herde de *SimAdapterInterface*, implementando o respectivo adaptador e sua porta. Dessa forma, estabelece-se um novo elo com a aplicação desejada, preservando o comportamento do controlador original, e promovendo maior modularidade e testabilidade ao sistema.

Desta maneira, a classe *NeuroFuzzyController* é a responsável por calcular as doses de insulina que serão retornadas ao simulador. Ela é tratada de maneiras diferentes em cada Prova de conceito.

Figura 19 – Diagrama de Classes da Segunda POC.



Fonte: Autor.

5.2.1 Prova de Conceito 1 - Lógica Difusa

O controlador de lógica difusa utiliza-se de três variáveis linguísticas: Glicose Atual, Taxa de Crescimento da Glicose e Aceleração da Glicose (MAUSETH et al., 2010), correspondendo, respectivamente, à unidade, à primeira derivada e à segunda derivada da glicose. Operacionalmente, no tempo t , os valores da taxa de crescimento e da aceleração são atualizados utilizando-se dos valores de CGM em t , $t - 1$ e $t - 2$ (MAUSETH et al., 2013). Assim, a Equação da taxa é dada por 5.1, e a Equação da aceleração por 5.2:

$$\text{taxa} = \frac{CGM_t - CGM_{t-1}}{\Delta t}; \quad (5.1)$$

$$\text{aceleração} = \frac{CGM_t - 2 \cdot CGM_{t-1} + CGM_{t-2}}{(\Delta t)^2}; \quad (5.2)$$

Além disso, a insulina deve ser administrada no paciente-teste a cada cinco minutos (MAUSETH et al., 2015), sendo, portanto, Δt igual a cinco.

Utilizou-se a técnica de Janela Deslizante de Tamanho Fixo para armazenar os elementos de forma conveniente. Essa técnica envolve mover uma janela de tamanho fixo através de uma estrutura de dados, geralmente uma matriz ou *string*, para resolver problemas de forma eficiente com base em subconjuntos contínuos de elementos (GE-

EKSFORGEEKS, 2017). Assim, todas as medições anteriores a $t - 2$ são descartadas, economizando memória a longo prazo.

Para implementar essa funcionalidade, utilizou-se uma classe abstrata chamada *Sliding_Window*, que realiza uma implementação genérica do algoritmo de janela deslizante. Essa classe é estendida pela classe *Glucose_Window*, baseada nos princípios do *Domain-Driven Design (DDD)*, e expressa no Código 1, implementando efetivamente as Equações 5.1 e 5.2.

Código 1 – Código da classe *Glucose_Window*.

```
1 class Glucose_Window(Sliding_Window):
2     def __init__(self, window_size, delta_t_minutes):
3         super().__init__(window_size)
4         self.delta_t_minutes = delta_t_minutes
5
6     def calculate_rate(self) → float:
7         return (self.data[-1] - self.data[-2]) / self.delta_t_minutes
8
9     def calculate_acc(self) → float:
10        return (self.data[-1] - 2 * self.data[-2] + self.data[-3]) /
11            (self.delta_t_minutes ** 2)
```

Fonte: Autor.

A classe *NeuroFuzzyController* realiza uma relação de composição com a classe *Glucose_Window*, armazenando as informações necessárias para a previsão dos valores de insulina. O tamanho da janela é de três unidades, correspondentes às últimas medições de glicose.

Dentro da classe *NeuroFuzzyController*, existe a função *calculate_dose*, que retorna um *float*. Tal função é responsável por calcular a dose que o usuário recebe no tempo t . Essa função recebe como parâmetros *current_glucose* (um *float*) e *current_time* (um *datetime*).

Inicialmente, o controlador verifica o horário da última medição. Caso ela tenha sido realizada há menos de cinco minutos, é retornado zero como dose de insulina, conforme ilustrado no Código 2.

Código 2 – Código do tempo de medição.

```
1 last_pump_diff_min: float = (current_time - self.last_pump).total_seconds() / 60
2 if last_pump_diff_min < TIME_MEASUREMENTS_MIN:
3     return 0
```

Fonte: Autor.

Caso o tempo desde a última medição seja adequado, os valores de glicose são inseridos na janela deslizante. No entanto, para que a taxa e a aceleração da glicose possam ser calculadas, o controlador precisa de ao menos três medições. Se esse critério

não for atendido, também é retornado zero como dose de insulina. Esse processo está demonstrado no Código 3.

Código 3 – Adição das medições de glicose.

```
1 self.registry.insert(current_glucose)
2 if not self.registry.is_valid():
3     return 0
```

Fonte: Autor.

Quando autorizado a calcular a dose, o controlador atualiza a variável *last_pump*, que armazena a última vez que a insulina foi infundida, e utiliza a variável *fuzzy_logic*, uma implementação da interface *FuzzyControllerInterface*, para calcular uma pré-dose baseada na lógica difusa. Esse processo é representado no Código 4.

Código 4 – Função para sugerir dose.

```
1 self.last_pump = current_time
2 fuzzy_insulin = self.fuzzy_logic.suggest_insulin(current_glucose,
3                                                    self.registry.calculate_rate(),
4                                                    self.registry.calculate_acc())
5
```

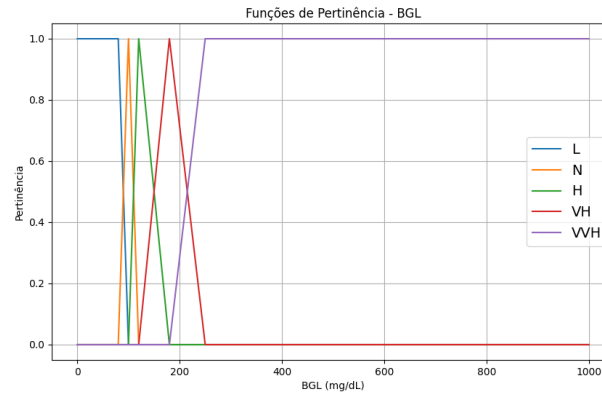
Fonte: Autor.

Para atribuir uma implementação concreta à variável *fuzzy_logic*, é utilizado o padrão de Injeção de Dependência, definindo a classe no momento da construção do *NeuroFuzzyController*. A implementação concreta é realizada pela classe *FuzzyController*. Essa interação representa, parcialmente, um adaptador secundário que isola a lógica difusa do restante da solução, promovendo a independência desse módulo.

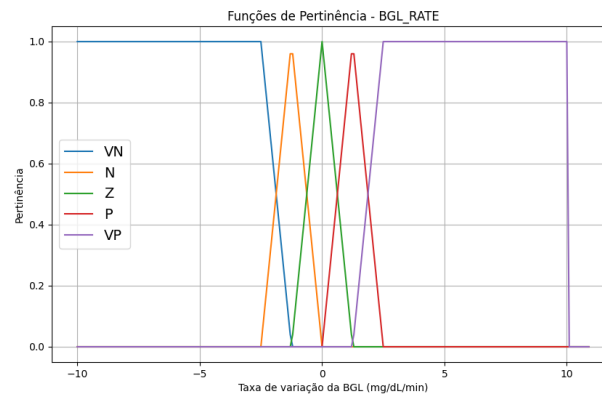
A classe *FuzzyController* utiliza a biblioteca *Python SkFuzzy* para realizar a previsão baseada em lógica difusa. A lógica difusa do controlador pode ser dividida em dois passos: definição dos Conjuntos Difusos e construção da Matriz de Dosagem.

5.2.1.1 Conjuntos Difusos

Para resolver um problema de lógica difusa, é necessário definir, inicialmente, as variáveis linguísticas e suas respectivas funções de pertinência. Como já evidenciado, utilizam-se três variáveis antecedentes: Glicose Atual (*BGL*), Taxa de Crescimento da Glicose (*BGL_RATE*) e Aceleração da Glicose (*BGL_ACC*). As Figuras 20 a 22 mostram as funções de pertinência das variáveis antecedentes.

Figura 20 – Função de pertinência da variável linguística *BGL*.

Fonte: Autor.

Figura 21 – Função de pertinência da variável linguística *BGL_RATE*.

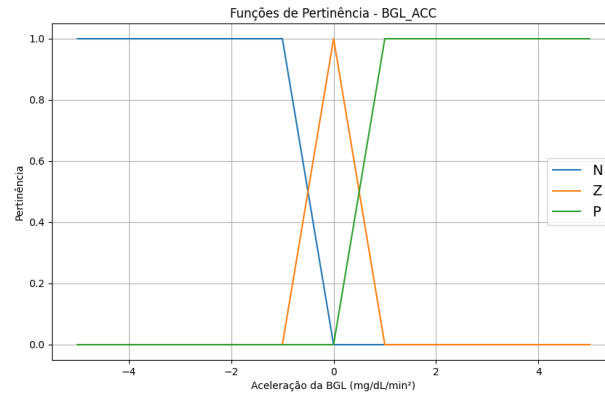
Fonte: Autor.

5.2.1.2 Matriz de Dosagem

Com as funções de pertinência definidas, é necessário estabelecer as regras difusas para prever a dose de insulina. A matriz de dosagem, desenvolvida por [Mauseth et al. \(2010\)](#), correlaciona as três variáveis antecedentes com a variável consequente ([MAUSETH et al., 2010](#)). A Figura 23 apresenta essa matriz atualizada conforme estudos desenvolvidos em 2015:

Destaca-se que os limites das funções de pertinência foram ajustados, pois os definidos originalmente causavam erros no simulador. Inicialmente, por exemplo, a variável *BGL* era limitada entre 40 e 400; no entanto, o simulador pode gerar valores fora dessa faixa. Assim, os limites foram estendidos para 0 e 1000 garantindo compatibilidade com todos os cenários. Ressalta-se que os foram definidos empiricamente, já que o simulador pode retornar valores compreendidos à esta faixa.

Utilizando-se dessa matriz, o método de inferência de Mamdani é aplicado para

Figura 22 – Função de pertinência da variável linguística *BGL_ACC*.

Fonte: Autor.

Figura 23 – Matriz de dosagem.

	Taxa de Glicose	VN			N			Z			P			VP		
	Aceleração da Glicose	N	Z	P	N	Z	P	N	Z	P	N	Z	P	N	Z	P
Glicose	VVH (180,250,400)	0.00	0.00	0.00	0.10	0.15	0.80	0.10	0.40	0.80	0.15	0.60	1.40	0.20	1.00	2.00
	VH (120,180,250)	0.00	0.00	0.00	0.05	0.10	0.25	0.10	0.30	0.40	0.10	0.50	1.20	0.15	0.80	1.90
	H (100,120,180)	0.00	0.00	0.00	0.00	0.05	0.10	0.05	0.20	0.30	0.10	0.30	1.00	0.10	0.50	1.80
	N (80,100,120)	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.10	0.20	0.05	0.20	0.45	0.05	0.20	0.60
	L (40,80,100)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.05	0.05	0.05	0.10

Fonte: (MAUSETH et al., 2015). Traduzido pelo autor.

determinar a dose de insulina via lógica difusa. Esse processo está representado no Código 5.

Código 5 – Código da Sugestão de Insulina.

```

1 def suggest_insulin(self, bgl:float, bgl_rate:float, bgl_acc:float) → float:
2     self.insulin_sim.input['BGL'] = bgl
3     self.insulin_sim.input['BGL_RATE'] = bgl_rate
4     self.insulin_sim.input['BGL_ACC'] = bgl_acc
5     self.insulin_sim.compute()
6
7     return round(self.insulin_sim.output['INSULIN'], 2)

```

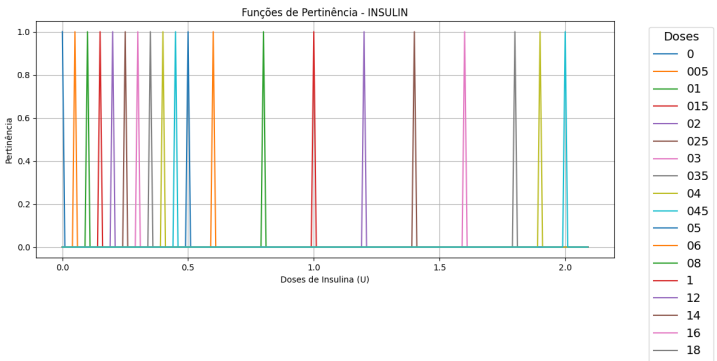
Fonte: Autor.

Assim, define-se também a função de pertinência da variável consequente *INSULIN*, ilustrada na Figura 24.

5.2.1.3 Fator de Personalização

Mauseth et al. (2013) propuseram o conceito do Fator de Personalização (*PF*), com o objetivo de individualizar a dose de insulina administrada a cada paciente, consi-

Figura 24 – Função de pertinência da variável linguística *INSULIN*.



Fonte: Autor.

derando sua necessidade diária, rotina e adesão ao tratamento. Com isso, a dosagem final é calculada pela Equação 5.3:

$$D_{\text{final}} = D_{\text{difusa}} \times TDD \times f_{\text{PF}}; \tag{5.3}$$

tem-se que D_{final} representa a dose final aplicada; D_{difusa} a dose base obtida da matriz de dosagem; TDD (*Total Daily Dose*) a insulina total diária requerida, e f_{PF} o fator escalar associado ao PF conforme a Tabela 4.

Tabela 4 – Relação entre Fator de Personalização e Fator Escalar

PF	Fator Escalar	Dose de Insulina	% em relação ao PF-5
0	0,032	1,74	190%
1	0,029	1,58	172%
2	0,026	1,41	154%
3	0,023	1,25	136%
4	0,020	1,08	118%
5	0,017	0,92	100%
6	0,014	0,75	82%
7	0,011	0,59	64%
8	0,008	0,42	46%
9	0,005	0,26	28%
10	0,002	0,09	10%

Fonte: (MAUSETH et al., 2013).Adaptado pelo Autor

Valores de PF mais baixos, como de exemplo o PF 0, correspondem a abordagens mais agressivas na administração de insulina; enquanto valores mais altos resultam em ajustes mais conservadores(MAUSETH et al., 2013). Vale ressaltar que todas as doses de insulina são da insulina do tipo *bolus* devido a sua rápida ação no paciente (SÍRIO-LIBANÊS, 2025).

Para a implementação do PF, basta definir uma matriz correspondente à Tabela 4 e aplicá-la na fórmula 5.3, utilizando o resultado para calcular a dose administrada ao usuário. No entanto, essa solução ainda apresenta algumas limitações, tais como a determinação do valor ideal de PF para cada paciente; a estimativa precisa do TDD (*Total Daily Dose*), e a definição da frequência e dos critérios para atualização dessas informações ao longo do tempo.

Diante dessas incertezas, foi proposta a utilização de técnicas de Inteligência Artificial, a fim de automatizar e aprimorar a personalização do tratamento.

5.2.2 Prova de Conceito 2 - Lógica Difusa e Inteligência Artificial

Para a Segunda Prova de Conceito, todo módulo de lógica difusa desenvolvido na Primeira Prova de Conceito foi mantido, exceto pela equação do fator de personalização, que foi substituída pela equação 5.4:

$$D_{\text{final}} = D_{\text{difusa}} \times PF_{\text{NN}}; \quad (5.4)$$

Tem-se que D_{final} representa a dose final aplicada; D_{difusa} corresponde à dose base obtida a partir da matriz de dosagem, e PF_{NN} é o fator escalar predito por uma [Rede Neural Recorrente](#). Dessa forma, elimina-se a necessidade de inserir manualmente o fator de personalização e o *Total Daily Dose* (TDD) do paciente, reduzindo a quantidade de informações necessárias para o funcionamento do pâncreas artificial.

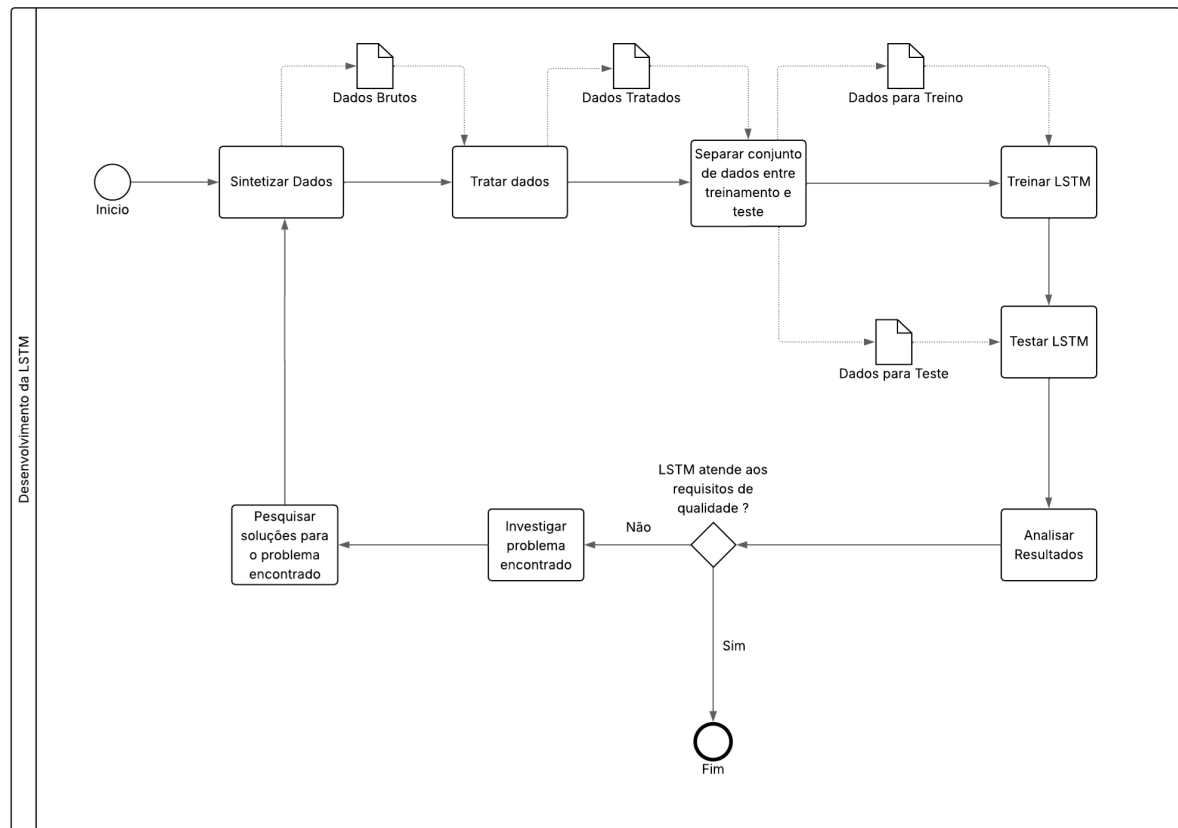
Uma propriedade relevante das LSTMs e das GRUs é a capacidade de ambos de lidarem eficientemente com dados sequenciais dependentes ao longo de períodos extensos, superando problemas de gradiente observados em outras arquiteturas de redes neurais recorrentes (RNNs). Assim, essa característica é explorada para, a partir do retrospecto do paciente, inferir o fator escalar mais adequado a ser utilizado em cada momento.

A Figura 25 representa um diagrama BPMN no qual descreve como foi realizado o desenvolvimento da Segunda Prova de Conceito. Com isso, o BPMN evidenciado da Figura 25 representa as fases que ocorreram no desenvolvimento das redes neurais da Segunda Prova de Conceito:

5.2.2.1 Sintetização dos Dados

Durante o desenvolvimento do pâncreas artificial por [Mauseth et al. \(2010\)](#), relatado em seu artigo, foi evidenciado que o fator de personalização (PF) pode ser ajustado dinamicamente em situações especiais, como períodos de alteração na sensibilidade à insulina, por exemplo em exercícios físicos, enfermidades ou menstruação. Essas modificações devem basear-se nos indicadores HBGI (*High Blood Glucose Index*) e LBGi (*Low Blood Glucose Index*).

Figura 25 – BPMN Desenvolvimento das Redes Neurais.



Fonte: Autor.

Segundo [Frontoni et al. \(2013\)](#), os índices HBGI e LBGI são indicadores preditivos que representam a frequência e a intensidade das medições elevadas e baixas de glicemia, respectivamente. Valores elevados de HBGI e LBGI indicam hiperglicemia e hipoglicemia mais frequentes ou mais severas. A Tabela 5 apresenta os níveis ideais e suas respectivas faixas de risco para ambos os índices.

Tabela 5 – Classificação dos níveis de risco segundo os valores de HBGI e LBGI

Nível de Risco	LBGI (Hipoglicemia)	HBGI (Hiperglicemia)
Mínimo	$\leq 1,1$	$\leq 5,5$
Baixo	1,1 – 2,5	5 – 10
Moderado	2,5 – 10	10 – 15
Alto	> 10	> 15

Fonte: ([FRONTONI et al., 2013](#)). Adaptado pelo Autor.

De acordo com [Mauseth et al. \(2010\)](#), os valores de HBGI e LBGI devem permanecer abaixo de 2,5, de forma que o risco de hipoglicemia esteja sempre abaixo do nível moderado, e o de hiperglicemia, no nível mínimo.

Com base nisso, para a sintetização dos dados, foi desenvolvida uma classe chamada `PFkick`, acoplada ao controlador, responsável por sugerir, em tempo real, um fator de personalização ideal com base nos valores atuais de HBGI e LBGI. Essa classe segue o padrão de projeto *Strategy*, o qual define uma família de algoritmos, encapsulando-os e tornando-os intercambiáveis, conforme descrito por [Gamma et al. \(1995\)](#). Dessa forma, a classe implementa as seguintes estratégias propostas pelo autor:

1. **Alteração dinâmica conforme os indicadores:** Se o HBGI ultrapassar o limite tolerável, o PF é aumentado; se o LBGI ultrapassar o tolerável, o PF é reduzido. Esta abordagem é implementada pela classe `change_by`.
2. **Alteração condicionada por limite de tentativas:** Aplica os mesmos princípios da estratégia anterior, mas com um número máximo de ajustes permitidos. Esta lógica é representada pela classe `change_max`.
3. **Sem alterações no PF:** O fator de personalização permanece constante, conforme definido na primeira prova de conceito. Esta estratégia é implementada pela classe `pf_naive`.

Todas as classes citadas estão disponíveis no repositório da monografia presente no seguinte link.¹

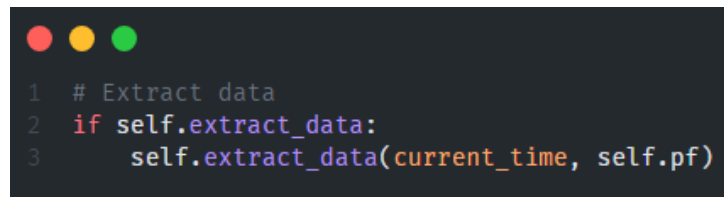
Para alimentar a rede neural com dados relevantes, foram utilizados os pacientes virtuais empregados na primeira prova de conceito. Para cada uma das estratégias descritas, a simulação foi executada 200 vezes, de modo a capturar uma diversidade significativa de padrões glicêmicos. As rotinas utilizadas foram aquelas relacionadas aos perfis de pacientes que serão descritos no [Capítulo 6](#).

A extração dos dados foi realizada por meio do método `extract_data`, implementado no controlador. O uso de tal método está evidenciado pela Figura 6. Esse método gera um arquivo `.csv` contendo os seguintes atributos: data e hora, última medição glicêmica, penúltima e antepenúltima medições, taxa de variação da glicose, aceleração da glicose, LBGI, HBGI, faixa etária (criança, adolescente ou adulto) e o PF, calculado como o produto do fator escalar original pelo TDD (*Total Daily Dose*).

Após a coleta, os dados foram pré-processados e utilizados no treinamento das redes neurais. Ao término do processo de síntese dos dados, determinaram-se as faixas da variável alvo para cada grupo etário, conforme apresentado na Tabela 6.

¹ Repositório da monografia: <https://github.com/kaua-pt/TCC_ArtificialPancreas>, último acesso em 06 de novembro de 2025.

Código 6 – Código da Extração de Dados do Simulador.



```

1 # Extract data
2 if self.extract_data:
3     self.extract_data(current_time, self.pf)

```

Fonte: Autor.

Tabela 6 – Faixa de Valores da Variável Alvo por Faixa Etária

Faixa Etária	Faixa da Variável Alvo (Unidades)
Crianças	0 até 0,4
Adolescentes	0 até 0,8
Adultos	0 até 1,2

Fonte: Autor.

5.2.2.2 Tratamento dos Dados

Para facilitar o desenvolvimento e promover maior organização, o processo de tratamento dos dados foi dividido em duas etapas principais:

1. **Processamento imediato após cada simulação:** consiste na execução de *scripts* de limpeza e validação dos dados logo após a finalização de cada rodada, garantindo a geração de *datasets* parciais organizados e compatíveis com as métricas estabelecidas, e
2. **Processamento consolidado ao final de todas as simulações:** etapa em que os *datasets* previamente limpos são integrados, reavaliados e preparados para o ambiente de treinamento da rede neural.

Essa divisão foi adotada considerando que o ambiente utilizado para o treinamento da rede neural foi o Google Colab, conforme detalhado no [Capítulo 3](#). Tal abordagem permitiu uma coleta mais sistemática e detalhada dos dados, além de proporcionar maior controle sobre a qualidade do conjunto final.

Ao final de cada rodada de simulação, era executado um notebook Jupyter denominado *clean_dataset*, responsável por realizar a primeira etapa do pré-processamento. Esse *script* segue os seguintes passos:

1. Leitura do último conjunto de dados gerado pela simulação;
2. Criação da coluna *Simulation_Number*, a qual orienta a segmentação dos dados e posterior divisão temporal (por dia) para o treinamento das redes neurais;

3. Aplicação das métricas definidas para avaliar o sucesso do pâncreas artificial, conforme estabelecido no [Capítulo 2](#), na seção de Sistemas de Controle;
4. Geração de um novo *dataset* contendo apenas os dias que atenderam aos critérios estabelecidos, com o objetivo de treinar a rede neural em dados representativos e de qualidade.

Todos os resultados e métricas obtidas foram armazenadas no repositório padrão do trabalho presente no seguinte link.²

É importante ressaltar que foram desenvolvidas três redes neurais, uma para cada faixa etária, com o objetivo de garantir uma melhor cobertura dos *outliers* principalmente para pacientes mais sensíveis, como é o caso das crianças e dos adolescentes. Sendo assim, utilizou-se um *script* em Python, denominado *merge_data*, responsável por unificar todos os conjuntos de dados válidos e compatíveis dentro de cada faixa etária.

Na etapa seguinte, já no ambiente do Google Colab e mais próximo do treinamento da rede neural, foi realizada uma nova rodada de tratamento de dados, com foco na preparação adequada para o aprendizado das redes neurais. Inicialmente, foi conduzida uma análise de correlação entre todas as variáveis por meio da correlação de Pearson. O objetivo dessa análise era identificar quais variáveis apresentavam forte relação com a variável alvo, permitindo a seleção das mais relevantes para o treinamento da IA. As matrizes de correlação geradas estão representadas pelas Figuras 26 a 28.

Com base nos resultados, para todas as redes neurais, foram removidas as colunas de *Simulation Number*, já que é irrelevante para o problema, *CGM2* e *CMG3* por representarem o mesmo dado, e por fim *Acceleration* por conter uma baixíssima correlação com a variável alvo. Dessa forma, o conjunto final de variáveis utilizadas foi reduzido a apenas quatro: Glicose Atual, Taxa de Crescimento da Glicose HBGI e LBGI.

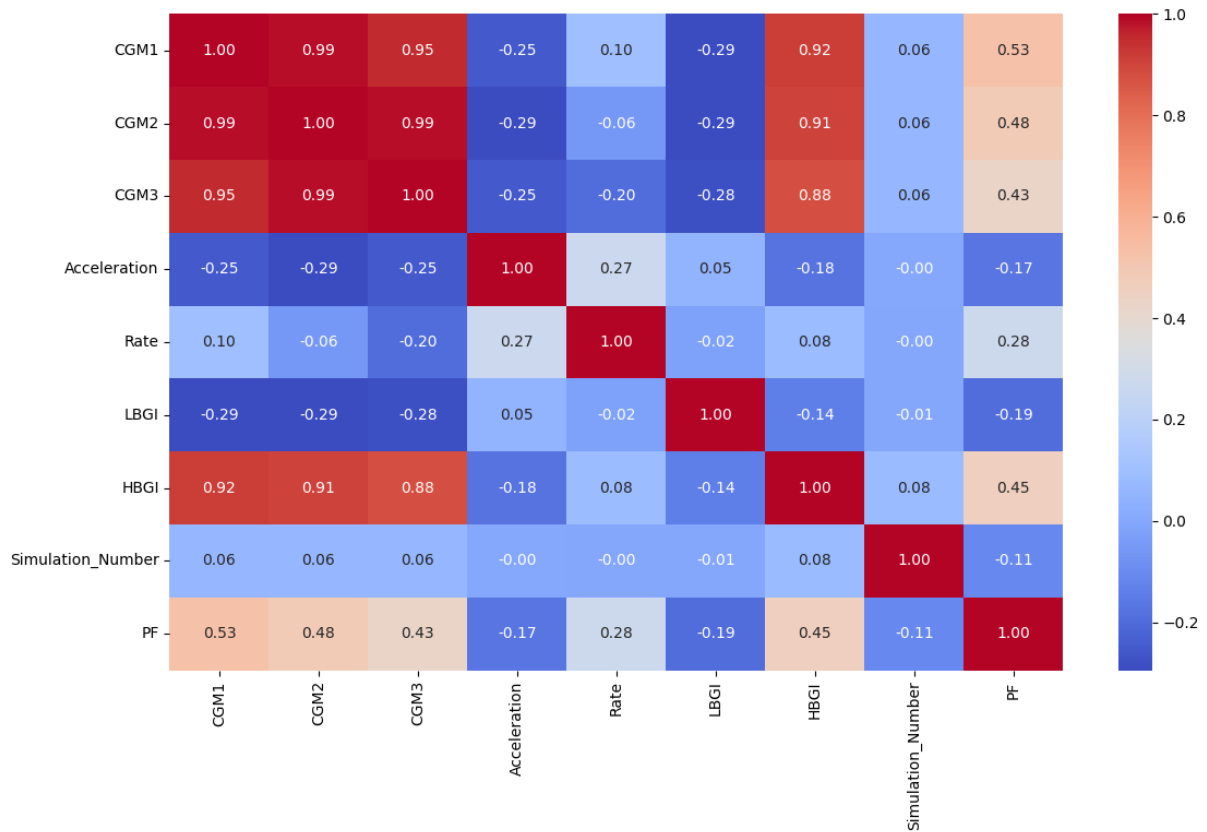
Por fim, foi desenvolvida uma função responsável por estruturar janelas temporais entre as simulações, permitindo que as redes neurais pudessem capturar dependências temporais e realizar uma melhor generalização do problema. O número de passos escolhido para a janela temporal encontra-se detalhado na Tabela 7. Esse valor foi selecionado com base em experimentos empíricos que demonstraram melhor desempenho para a faixa etária analisada. Além disso, priorizou-se o uso de janelas menores com o intuito de intensificar a capacidade de generalização da rede neural.

5.2.2.3 Treinamento das Redes Neurais

Para efetivamente treinar as Redes Neurais, os *datasets* foram separados em três conjuntos distintos:

² Repositório do trabalho: <https://github.com/kaue-pt/TCC_ArtificialPancreas/tree/main/docs/cleandatasets>, último acesso em 06 de novembro de 2025.

Figura 26 – Matriz de Correlação - Crianças.



Fonte: Autor.

Tabela 7 – Quantidade de Passos de Janela Temporal Por Faixa Etária

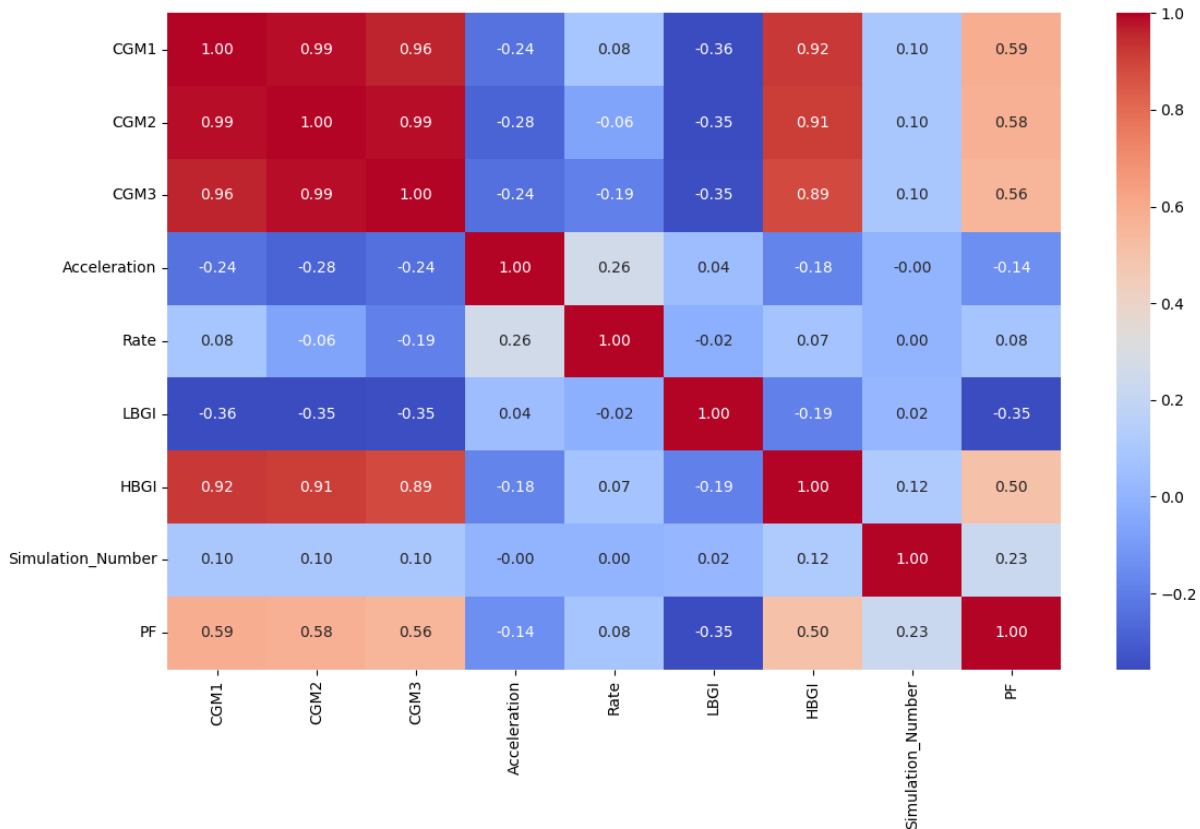
Faixa Etária	Tamanho da Janela Temporal
Adulto	16
Adolescente	16
Criança	6

Fonte: Autor.

1. **Treino (60%):** utilizado para o ajuste dos pesos da rede neural;
2. **Validação (20%):** utilizado para monitorar o desempenho da rede durante o treinamento e prevenir *overfitting*, e
3. **Teste (20%):** utilizado para avaliar o desempenho final do modelo treinado em dados nunca vistos.

Cada *dataset* foi submetido a diferentes arquiteturas de redes neurais, com o objetivo de identificar aquela com melhor desempenho para o problema em estudo. Todas

Figura 27 – Matriz de Correlação - Adolescentes.



Fonte: Autor.

as tentativas foram documentadas e estão disponíveis no seguinte link.³

Em todas as execuções, as métricas utilizadas para avaliação foram o *Mean Absolute Error (MAE)* e o *Root Mean Squared Error (RMSE)*. Entre as arquiteturas testadas, destacam-se aquelas que apresentaram os melhores resultados, descritas a seguir, para cada caso: Crianças, Adolescentes e Adultos.

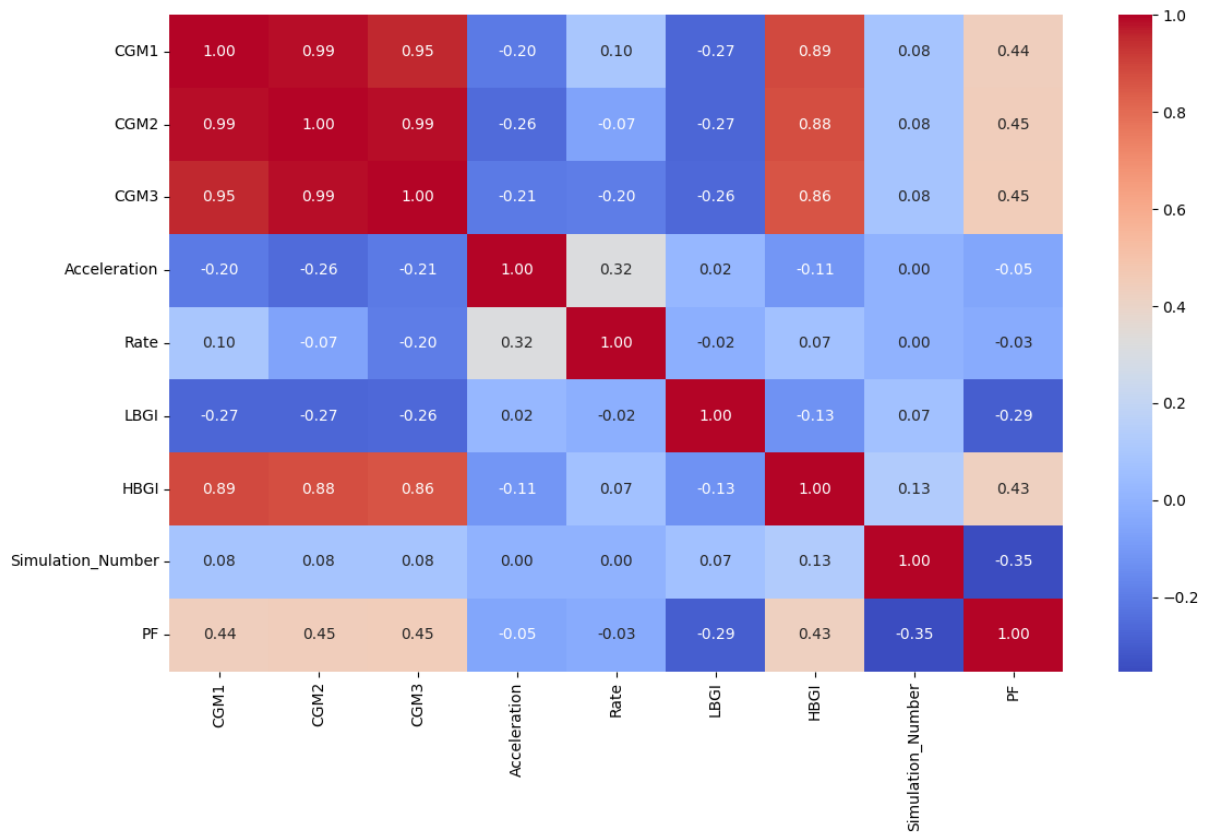
5.2.2.3.1 Crianças

Para o grupo de crianças, a arquitetura de rede neural com melhor desempenho foi uma *Attention GRU*, composta por uma camada de GRU seguida de um mecanismo de atenção, permitindo que a rede confira maior ênfase a partes relevantes da sequência e capture variações importantes nos dados.

Inicialmente, os dados passam por uma camada de *GRU* com 128 neurônios, seguida do mecanismo de atenção disponibilizado pelo próprio *Keras*. A saída é então re-

³ Repositório do projeto: <https://github.com/kaau-pt/TCC_ArtificialPancreas/tree/main/docs>, último acesso em 06 de novembro de 2025.

Figura 28 – Matriz de Correlação - Adultos.



Fonte: Autor.

duzida por meio de uma camada de *Global Average Pooling*, resultando em um vetor de características representativo da sequência.

Em seguida, esse vetor é processado por uma camada de *Dropout* com taxa de 20% e por duas camadas densas com 64 e 32 neurônios, ambas utilizando função de ativação *ReLU*. A camada de saída contém apenas um neurônio responsável pela predição final da variável alvo.

O modelo foi treinado com o otimizador *Adam*, configurado com taxa de aprendizado de 10^{-4} , utilizando como função de perda o *Mean Squared Error (MSE)*.

Durante o treinamento, foram empregados os seguintes *callbacks*:

1. **EarlyStopping**: interrompe o treinamento caso a Raiz do Erro Quadrático Médio de validação, aqui tratada pela sigla RMSE, não apresente melhora superior a 10^{-3} em 15 épocas consecutivas, restaurando automaticamente os melhores pesos obtidos;
2. **ReduceLROnPlateau**: reduz a taxa de aprendizado em um fator de 0,1 sempre que o RMSE de validação não melhora por 3 épocas seguidas, com taxa mínima de 10^{-8} e *cooldown* de 2 épocas, promovendo melhor convergência, e

3. **ModelCheckpoint**: salva o modelo ao final de cada época, monitorando o RMSE de validação e permitindo manter versões consistentes do modelo ao longo do treinamento.

O treinamento foi realizado por até 200 épocas, com *batch size* de 2048 amostras.

5.2.2.3.2 Adolescentes

Para o grupo de adolescentes, a arquitetura de rede neural com melhor desempenho foi baseada em uma *Bidirectional LSTM*.

Inicialmente, os dados passam por uma camada de *Bidirectional LSTM* com 128 unidades, configurada para retornar sequências completas, permitindo que o modelo aprenda dependências tanto do passado; quanto do futuro em relação a cada ponto da sequência. Em seguida, é aplicada uma segunda camada de *LSTM* com 64 neurônios, também retornando sequências, com a finalidade de refinar as representações temporais. Posteriormente, os dados passam por uma terceira camada de *LSTM* com 32 neurônios, desta vez sem retorno de sequência.

Na etapa densa, o modelo é composto por uma camada *Dense* com 64 neurônios e ativação ReLU, seguida de outra camada densa com 32 neurônios e a mesma função de ativação. A camada de saída contém apenas um neurônio responsável pela predição final da variável alvo.

Após cada camada de *LSTM*, foram adicionadas camadas de *Batch Normalization* e *Dropout*, com taxas de 30%, 20% e 20%, respectivamente. Ao final, o modelo passa para a sua etapa densa, composta por uma camada *Dense* com 64 neurônios, e ativação ReLU, seguida por outra camada densa de 32 neurônios com a mesma função de ativação, para assim, finalmente, a camada de saída ser representada por um neurônio contendo a previsão final para a variável alvo.

O treinamento é realizado com o otimizador Adam, configurado com taxa de aprendizado inicial de $5 * 10^{-4}$, parâmetros $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$, e a função de perda escolhida foi o *Huber Loss* ($\delta = 0.1$) função que garante mais robustez em casos de *outliers*.

O treinamento foi realizado por até 100 épocas, com *batch size* de 32 amostras, empregando os mesmos *callbacks* descritos no modelo para crianças.

5.2.2.3.3 Adultos

Primeiramente, os dados de entrada passam por uma camada *LSTM* com 64 neurônios, configurada para retornar sequências. Esta camada utiliza regularização L2 ($l2 = 10^{-3}$) para reduzir o risco de *overfitting* e aplica *recurrent dropout* de 10%, que

desliga aleatoriamente algumas conexões recorrentes durante o treinamento, melhorando a generalização.

Em seguida, é aplicada uma camada de *Dropout* com taxa de 30%. Depois, os dados passam por uma segunda camada *LSTM* com 32 neurônios, também com regularização L2 ($l2 = 10^{-2}$).

O vetor de saída é então processado por uma camada *Dense* com 16 neurônios e ativação ReLU, com regularização L2. Por fim, a camada de saída contém apenas um neurônio, que gera a predição da variável alvo.

O treinamento dos adultos utilizou-se das mesmas configurações do treinamento para os adolescentes quanto em relação às métricas, *callbacks* e hiperparâmetros.

Ao final do treinamento, cada rede neural gerou um arquivo *.keras*, no qual é o modelo treinado, e um arquivo *.joblib*, que foi utilizado para a normalização dos dados no processo de predição.

5.2.2.4 Análise das Métricas

Por fim, após a finalização do treinamento dos modelos, foi realizado um teste para validar a eficácia das redes neurais e avaliar se ocorreu *overfitting* ou *underfitting*. As melhores métricas obtidas estão apresentadas na Tabela 8.

Tabela 8 – Desempenho do modelo por faixa etária

Faixa Etária	MAE	RMSE	R ²
Crianças	0,0306	0,0396	0,6617
Adolescentes	0,0547	0,0910	0,8064
Adultos	0,0220	0,0442	0,8617

Fonte: Autor.

Para uma melhor compreensão das métricas, é necessário considerar a faixa de valores da variável alvo em cada faixa etária apresentada ao final da seção de [Sintetização de Dados](#). Com isso, o erro relativo em porcentagem das métricas *Mean Absolute Error* e *Root Mean Square Error* foi calculado utilizando a Equação 5.5.

$$\text{Erro relativo (\%)} = \frac{\text{MAE ou RMSE}}{\text{range}} \times 100; \quad (5.5)$$

Os resultados mostram que:

- **Crianças:** MAE relativo de 7,65% e RMSE relativo de 9,90%. Apesar do R² mais baixo, o erro proporcional à faixa da variável alvo não é elevado;

- **Adultos:** Excelente desempenho, MAE relativo de 1,83% e RMSE relativo de 3,68%, demonstrando que o modelo apresentou o melhor desempenho proporcional, refletido também no maior R^2 , e
- **Adolescentes:** MAE relativo de 6,84% e RMSE relativo de 11,38%, indicando maior dispersão dos dados nesta faixa etária, possivelmente devido à variabilidade fisiológica característica do crescimento.

Portanto, ao analisar o erro em relação à faixa de valores da variável alvo, observa-se que os modelos conseguem fornecer previsões razoavelmente precisas, mesmo para faixas etárias com maior variabilidade fisiológica, como crianças e adolescentes.

5.2.2.5 Utilização do Modelo

Com a finalização dos treinamentos, foi desenvolvida a classe *PfController*, responsável por realizar o controle do PF por meio da rede neural. Naturalmente, essa classe, além de atuar como um adaptador secundário, também desempenha o papel de *strategy*, uma vez que a escolha do modelo e suas respectivas estratégias variam conforme a faixa etária do paciente.

Dessa forma, foram implementadas três novas classes com comportamentos semelhantes: *AdultStrategy*, *AdolescentStrategy* e *ChildStrategy*. Cada uma delas é responsável por carregar, a partir de variáveis de ambiente, o caminho do modelo, o arquivo de normalização correspondente e a janela de passos associada à faixa etária.

Durante a inicialização do controlador, a classe compatível com a faixa etária do paciente é instanciada. Em seguida, é criada uma janela deslizante, responsável por armazenar as últimas entradas dos índices HBGI e LBGI, da Taxa de Crescimento da Glicose e da Glicose Atual. Esses valores são repassados à rede neural para que seja realizada a predição do PF.

Vale destacar que, caso o número de predições anteriores seja inferior ao tamanho da janela, os valores ausentes são preenchidos com zero, garantindo a integridade da entrada fornecida à rede neural.

Um comportamento diferente ocorreu na classe *AdultStrategy*. O modelo correspondente estava prevendo valores ligeiramente superiores ao necessário para o controle da glicose do paciente. Dessa forma, todo resultado fornecido pelo modelo foi multiplicado por 0,60, valor obtido empiricamente. O mesmo ocorreu na classe *AdolescentStrategy*, na qual todos os valores de saída foram multiplicado por 10.

5.3 Resumo do Capítulo

Neste capítulo, foram abordados conceitos fundamentais sobre a diabetes tipo 1, os quais são essenciais para o desenvolvimento das provas de conceito deste trabalho. Inicialmente, foi apresentada uma base arquitetural para as provas de conceito, a qual se revelou como uma [Arquitetura de Portas e Adaptadores](#). Em seguida, foi discutido como construir um pâncreas artificial utilizando exclusivamente [Lógica Difusa](#), e como seria possível unificar [Inteligência Artificial](#) e [Lógica Difusa](#) para tal desenvolvimento.

No primeiro caso, foram aplicadas técnicas como a janela deslizante para armazenar medições históricas de glicose. Quando permitido pelo controlador, essas medições são utilizadas para processar uma dose de glicose, que é então aplicada ao paciente após um pós-processamento. Esse processo envolve uma fórmula que incorpora um fator de personalização, sendo esse o produto de um fator escalar, da dose fornecida pela [Lógica Difusa](#) e da quantidade diária de insulina necessária ao paciente.

No segundo caso, com o objetivo de evitar a necessidade de ajustes constantes no fator de personalização, foi empregada uma rede neural para lembrar informações do usuário ao longo das iterações, permitindo que o modelo retornasse um fator de personalização ajustado de acordo com as necessidades específicas do paciente em cada momento.

6 Análise de Resultados

Este capítulo apresenta os resultados obtidos pelo Pâncreas Artificial. Inicialmente, descreve-se o [Plano de Ação](#) adotado para a análise dos resultados deste trabalho. Em seguida, é apresentada a [Coleta de Dados](#), contendo as informações obtidas na [Primeira Prova de Conceito](#) e na [Segunda Prova de Conceito](#). Posteriormente, realiza-se a [Avaliação do Plano de Execução](#) proposto. Por fim, apresenta-se o [Resumo do Capítulo](#).

O método de análise de resultados, acordado no [Capítulo 4 - Metodologia](#) (seção 4.5), estabelece as seguintes fases de pesquisa-ação: Definição do Problema; Pesquisa Preliminar, Formulação de Hipóteses; Desenvolvimento do Plano de Ação; Implementação do Plano de Ação; Coleta de Dados; Avaliação do Plano de Execução, e Comunicação dos Resultados.

As primeiras três fases foram concluídas ainda na primeira etapa desse trabalho, com destaque para: Definição do Problema: esse detalhamento foi descrito na [seção 1.1 do Capítulo 1 - Introdução](#), e justificado na [seção 1.3](#) do mesmo capítulo, resultando no objeto de estudo desse trabalho, sendo esse um pâncreas Artificial para Diabéticos Tipo 1 com Lógica Difusa e Aprendizado Profundo;

Pesquisa Preliminar: essa pesquisa foi realizada via levantamento bibliográfico, descrito na [seção 4.2 do Capítulo 4 - Metodologia](#), e resultando na documentação dos principais referenciais teóricos e tecnológicos, respectivamente, revelados no [Capítulo 2 - Referencial Teórico](#) e [Capítulo 3 - Suporte Tecnológico](#), e

Formulação de Hipóteses: sumarizada na [Questão de Pesquisa, seção 1.2 do Capítulo 1 - Introdução](#), sendo "É possível desenvolver, in silico, um sistema de controle de glicose que proporcione maior comodidade ao usuário, minimizando a necessidade de inserção manual de informações pessoais?". Assumi-se como hipótese central, nesse caso, que o pâncreas artificial proporcionará maior comodidade ao usuário minimizando a necessidade de inserção manual de informações pessoais.

As fases de Desenvolvimento do Plano de Ação e Implementação do Plano de Ação são tratadas em Plano de Ação. A fase de Coleta de Dados tem seção dedicada. Já as fases Avaliação do Plano de Execução e Comunicação dos Resultados são apresentadas de forma conjunta em Avaliação do Plano de Execução.

6.1 Plano de Ação

Para a valoração das Provas de Conceito, elaborou-se um plano de ação baseado na submissão do controlador ao simulador [SimGlucose](#). Essa ferramenta disponibiliza

aproximadamente trinta pacientes virtuais, dez crianças, dez adolescentes e dez adultos, cada um apresentando respostas distintas à insulina e aos carboidratos.

O principal objetivo dos testes é verificar se as metas definidas no [Capítulo 2 - Referencial Teórico \(Seção 2.1\)](#) estão sendo atendidas pelo controlador. Um aspecto relevante considerado na análise é o erro aleatório introduzido pelo simulador nas medições do CGM, o que pode afetar a precisão dos resultados. Por esse motivo, cada simulação foi repetida dez vezes por paciente e por tipo de teste. Este número foi escolhido pois, após aproximadamente dez execuções, o cenário apresenta convergência estável nos valores de glicemia simulados, reduzindo de forma significativa as flutuações estatísticas inerentes ao domínio. Em seguida, foram calculados a média e o desvio-padrão de cada conjunto, de modo a avaliar de forma mais robusta a eficácia do controlador.

Para a realização dos experimentos, foi necessário desenvolver *Personas* que representassem os pacientes-teste, bem como definir as rotinas às quais cada uma delas seria submetida.

6.1.1 Personas e Rotinas

Em paralelo ao desenvolvimento das provas de conceito, foram definidas personas com base nos artigos utilizados como referência. A intenção dessa especificação é esclarecer sobre o público alvo desse trabalho. Foram estabelecidas três personas principais:

- **Persona 1:** Homem, 35 anos. Apresenta receio de episódios de hipoglicemia e deseja controle rigoroso, embora os tolere relativamente bem. Mantém uma rotina de esportivo e pratica exercícios físicos ao meio-dia. Seus fins de semana seguem o mesmo padrão ([MAUSETH et al., 2010](#));
- **Persona 2:** Menina, 6 anos. Não tem consciência do que é hipoglicemia. Frequenta a escola cinco dias por semana sem acompanhamento de uma enfermeira. Seus pais demonstram preocupação com o controle glicêmico. Sua alimentação é irregular, e ela participa de aulas de basquete e dança três vezes por semana. Ambos os perfis de aulas nos mesmos dias da semana([MAUSETH et al., 2010](#)); e
- **Persona 3:** Adolescente, 14 anos, do sexo masculino. Mantém uma rotina relativamente estruturada, seguindo horários fixos para refeições e monitoramento da glicose via CGM. Está comprometido com o controle glicêmico, e segue com disciplina os planos alimentares e as orientações médicas. Ainda manifesta receio de episódios de hipoglicemia, especialmente fora de casa, mas realiza ajustes com base no monitoramento contínuo e no apoio da família, baseado em [Mauseth et al. \(2010\)](#).

O protocolo alimentar padrão consiste em três refeições diárias: café da manhã com 45g de carboidratos, almoço com 70g de carboidratos e jantar com 80g de carboidratos,

aplicando-se a todos os três grupos etários (MAUSETH et al., 2010). Além disso, foram adicionadas outras rotinas específicas para tentar representar o dia a dia de cada persona, conforme as descrições.

Para a avaliação da qualidade de cada controlador nos diferentes cenários, serão utilizadas as métricas definidas no Capítulo 2 - Seção 2.1.

6.1.1.1 Testes de Estresse

Além do protocolo padrão, foram implementados testes de estresse com o objetivo de avaliar a robustez das estratégias de controle. Cada um desses testes foi projetado para evidenciar um fator específico:

- **Estresse Sem Alimentação:** Ausência de ingestão de alimentos durante o período de testes. O controlador deve evitar tanto hiperglicemia; quanto hipoglicemia;
- **Estresse por Exercício:** Ingestão do café da manhã habitual às 8h; 60g de carboidratos às 11h, 75g no jantar e 20g antes de dormir. O controlador deve evitar que o paciente entre em hipoglicemia(MAUSETH et al., 2015), e
- **Estresse por Alimentação:** Ingestão de 120 g de carboidratos às 11h. O controlador deve evitar que o paciente entre em hiperglicemia(MAUSETH et al., 2015).

Dessa forma, para cada rotina alimentar, foram realizadas pequenas alterações arbitrárias com a finalidade de capturar nuances da realidade. Apresentam-se duas tabelas: a Tabela 9, que apresenta a agenda alimentar padrão de cada cenário, e a Tabela 10 para cenários específicos, levando em consideração a descrição de cada persona.

Cada cenário é representado pela letra C seguida do número correspondente. Os cenários C1 até C4 representam as rotinas alimentares padrão; enquanto os cenários C5 a C7 referem-se aos testes de estresse.

Nos testes de estresse, adotaram-se os seguintes limites de referência: valores de glicemia inferiores a 60 mg/dL foram considerados como episódios de hipoglicemia, enquanto valores superiores a 300 mg/dL foram classificados como hiperglicemia. Esses limiares foram definidos com base nos critérios utilizados por Mauseth et al. (2013).

Na Tabela 10, são apresentados os cenários C8 a C10, cada um correspondente a uma persona, identificada pela letra P, ordenadas de acordo com a enumeração estabelecida no início da seção. Para cada cenário, foram desenvolvidas três rotinas distintas, alinhadas às características e necessidades específicas de cada persona, descritas anteriormente.

Tabela 9 – Agenda Alimentar Padrão por Horário e Cenário com Quantidade de Carboidrato (g).

Hora	C1	C2	C3	C4	C5	C6	C7
00:00	–	–	–	–	–	–	–
07:00	45	–	–	40	–	–	–
08:00	–	45	–	–	–	45	–
08:30	–	–	45	–	–	–	–
10:00	–	–	–	30	–	–	–
11:00	–	60	–	–	–	60	120
11:30	–	–	50	–	–	–	–
12:00	70	–	–	–	–	–	–
13:00	–	–	–	60	–	–	–
14:00	–	15	–	–	–	–	–
14:30	–	–	25	–	–	–	–
16:00	–	–	–	20	–	–	–
18:00	80	75	50	–	–	–	–
19:00	–	–	–	–	–	75	–
19:30	–	–	–	50	–	–	–
21:00	–	–	15	–	–	–	–
22:00	–	20	–	15	–	20	–
23:00	–	–	–	–	–	–	–

Fonte: Autor.

6.2 Coleta de Dados

Nesta seção, são apresentados os dados coletados referentes à [Primeira Prova de Conceito](#) e à [Segunda Prova de Conceito](#) deste estudo.

Para a análise dos resultados, foi desenvolvido um *script* em *Python* responsável pelo processamento dos dados retornados pelo simulador, com o objetivo de verificar se o controlador atingiu as metas previamente estabelecidas. Além disso, o *script* gera informações complementares, como os valores máximo e mínimo de glicose registrados, bem como seus respectivos desvios-padrão.

Dessa forma, as próximas subseções apresentam os pacientes utilizados nos testes *in silico* e os resultados obtidos a partir das simulações.

6.2.1 Resultados da Primeira Prova de Conceito

Os testes da Primeira Prova de Conceito foram realizados em três pacientes arbitrários de cada perfil, utilizando os [Pacientes-Teste](#). O autor realizou ajustes nos valores de TDD e no fator de personalização de cada paciente, afim de encontrar os parâmetros mais adequados para cada indivíduo.

Tabela 10 – Agenda Alimentar Específica por Horário e Cenário com Quantidade de CHO (g).

Hora	C8-P1	C9-P1	C10-P1	C8-P2	C9-P2	C10-P2	C8-P3	C9-P3	C10-P3
06:00	–	–	–	–	–	–	–	–	–
07:00	–	45	35	35	35	–	45	35	–
07:30	40	–	–	–	–	–	–	–	–
08:00	–	–	–	–	–	45	–	–	40
09:30	–	–	10	15	–	–	–	–	–
10:00	10	10	–	–	10	10	10	20	–
11:00	–	–	–	–	–	–	–	–	15
11:30	20	20	15	20	–	–	–	–	–
12:00	–	–	–	–	15	–	–	–	–
12:30	–	–	–	–	–	–	–	–	–
13:00	60	70	80	40	50	50	50	45	–
13:30	–	–	–	–	–	–	–	–	–
14:00	–	–	–	–	–	–	–	–	55
15:30	–	–	–	–	–	–	–	10	–
16:00	–	–	–	20	–	15	15	–	–
17:00	–	–	–	–	–	–	–	–	20
18:00	70	–	60	45	50	–	–	20	–
18:30	–	40	–	–	–	55	–	–	–
19:00	–	–	–	–	–	–	50	–	–
19:30	–	–	–	–	–	–	–	–	–
20:00	–	–	–	–	–	–	–	45	45
21:00	10	5	10	10	10	10	–	–	–
22:00	–	–	–	–	–	–	15	–	–
23:00	–	–	–	–	–	–	–	10	15

Fonte: Autor.

6.2.1.1 Crianças

Para realizar os testes, foram escolhidas, arbitrariamente, as crianças 1, 4 e 7. Com isso, conforme a Tabela 11, foram definidos os seguintes valores de TDD e PF:

Tabela 11 – Valores de TDD e PF para Diferentes Crianças Simuladas.

Criança	TDD	PF
1	12.35	8
4	12.35	8
7	12.65	5

Fonte: Autor.

Os resultados obtidos pela criança 1 estão apresentados na Tabela 12:

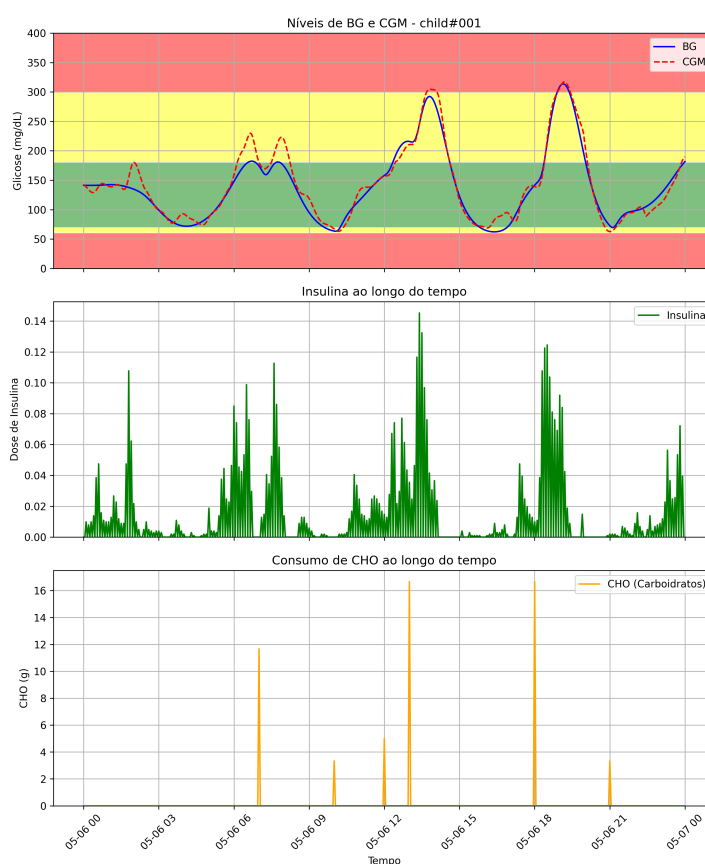
Nota-se que a única métrica não atingida foi o coeficiente de variação. Isso ocorre devido à rotina desbalanceada representada pelo cenário 1, no qual provoca picos de glicose, seguidos por uma infusão excessiva de insulina, ocasionando quedas acentuadas nos níveis glicêmicos e, conseqüentemente, grande variação. Essa oscilação pode ser observada na Figura 29.

Tabela 12 – Resultados da Primeira Prova de Conceito: Criança 1.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	70.29	9.13
Maior Glicose (mg/dL)	324.28	44.38
Porcentagem no Alvo (70–180 mg/dL) (%)	76.30	4.12
Hipo Nível 1 (54–69 mg/dL) (%)	1.98	2.77
Hipo Nível 2 (<54 mg/dL) (%)	0.05	0.29
Coefficiente de Variação (%)	39.68	7.51

Fonte: Autor.

Figura 29 – Gráfico de Glicose da Criança 1.



Fonte: Autor.

Nos testes de estresse, a criança 1 obteve os seguintes resultados, apresentados na Tabela 13:

A criança 4 atingiu todas as metas, conforme mostra a Tabela 14:

Pode-se observar que o paciente atingiu as metas. No entanto, ao se considerar o erro padrão, o coeficiente de variação ainda se encontra fora do ideal. Isso ocorre, assim como no caso da criança 1, devido à inadequada rotina alimentícia.

Nos testes de estresse, a criança 4 apresentou os resultados descritos na Tabela 15.

Tabela 13 – Resultados dos Testes de Estresse da Criança 1 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	74.63 $\pm$ 12.29	64.66 $\pm$ 12.52	64.36 $\pm$ 18.07
Maior Glicose (mg/dL)	238.99 $\pm$ 93.52	318.04 $\pm$ 85.46	332.38 $\pm$ 108.95
Hipoglicemia (%)	10	30	40
Hiperglicemia (%)	10	10	30

Fonte: Autor.

Tabela 14 – Resultados da Primeira Prova de Conceito: Criança 4.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	78.22	12.22
Maior Glicose (mg/dL)	275.39	37.17
Porcentagem no Alvo (70–180 mg/dL) (%)	77.13	3.29
Hipo Nível 1 (54–69 mg/dL) (%)	1.02	2.02
Hipo Nível 2 (<54 mg/dL) (%)	0.06	0.29
Coefficiente de Variação (%)	31.29	7.94

Fonte: Autor.

Tabela 15 – Resultados dos Testes de Estresse da Criança 4 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	89.00 $\pm$ 18.62	76.71 $\pm$ 12.59	88.01 $\pm$ 13.06
Maior Glicose (mg/dL)	184.63 $\pm$ 23.11	271.57 $\pm$ 18.25	393.81 $\pm$ 30.61
Hipoglicemia (%)	10	10	0
Hiperglicemia (%)	0	0	50

Fonte: Autor.

A criança 7 apresentou hipoglicemia de nível 1. Além disso, conseguiu atingir todas as metas estabelecidas. A Tabela 16 mostra seus resultados.

Tabela 16 – Resultados da Primeira Prova de Conceito: Criança 7.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	85.21	9.97
Maior Glicose (mg/dL)	200.90	20.14
Porcentagem no Alvo (70–180 mg/dL) (%)	92.72	5.03
Hipo Nível 1 (54–69 mg/dL) (%)	0.07	0.30
Hipo Nível 2 (<54 mg/dL) (%)	0.00	0.00
Coefficiente de Variação (%)	22.03	4.03

Fonte: Autor.

Nos testes de estresse, a criança 7 obteve os resultados apresentados na Tabela 17.

Tabela 17 – Resultados dos Testes de Estresse da Criança 7 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	78.65 $\pm$ 7.13	65.04 $\pm$ 8.13	47.62 $\pm$ 3.16
Maior Glicose (mg/dL)	134.96 $\pm$ 2.25	257.36 $\pm$ 9.70	286.94 $\pm$ 16.75
Hipoglicemia (%)	0	10	100
Hiperglicemia (%)	0	0	0

Fonte: Autor.

6.2.1.2 Adolescentes

Para realizar os testes, foram selecionados, arbitrariamente, os Adolescentes 2, 5 e 8. Conforme a Tabela 18, os valores de TDD e PF utilizados para cada um estão apresentados a seguir.

Tabela 18 – Valores de TDD e PF para Diferentes Adolescentes Simulados.

Adolescente	TDD	PF
2	27	5
5	14.6	5
8	27	5

Fonte: Autor.

O Adolescente 2 conseguiu atingir suas metas glicêmicas, como evidenciado na Tabela 19.

Tabela 19 – Resultados da Primeira Prova de Conceito: Adolescente 2.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	81.15	12.98
Maior Glicose (mg/dL)	239.53	19.98
Porcentagem no Alvo (70–180 mg/dL) (%)	69.69	6.12
Hipo Nível 1 (54–69 mg/dL) (%)	0.44	1.06
Hipo Nível 2 (<54 mg/dL) (%)	0.04	0.17
Coefficiente de Variação (%)	27.90	4.11

Fonte: Autor.

Nos testes de estresse, os resultados do Adolescente 2 estão apresentados na Tabela 20.

O Adolescente 2 apresentou episódios de hipoglicemia em todos os casos dos cenários 6 e 7, além de registrar hiperglicemia em oito ocasiões no cenário 7. Aliado aos resultados dos testes-padrão, pode-se inferir que o paciente necessita de uma rotina alimentar mais equilibrada para manter as metas de glicemia.

Tabela 20 – Resultados dos Testes de Estresse do Adolescente 2 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	78.27 $\pm$ 8.14	14.96 $\pm$ 8.48	2.20 $\pm$ 4.64
Maior Glicose (mg/dL)	152.42 $\pm$ 0.01	284.33 $\pm$ 8.12	413.25 $\pm$ 15.38
Hipoglicemia (%)	0	100	100
Hiperglicemia (%)	0	0	80

Fonte: Autor.

O Adolescente 5 conseguiu atingir as metas estabelecidas, conforme mostrado na Tabela 21.

Tabela 21 – Resultados da Primeira Prova de Conceito: Adolescente 5.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	82.53	12.61
Maior Glicose (mg/dL)	281.12	41.24
Porcentagem no Alvo (70–180 mg/dL) (%)	73.64	6.52
Hipo Nível 1 (54–69 mg/dL) (%)	1.81	4.44
Hipo Nível 2 (<54 mg/dL) (%)	0.04	0.35
Coefficiente de Variação (%)	34.92	8.08

Fonte: Autor.

A Tabela 22 ilustra os resultados obtidos por cenário, destacando um coeficiente de variação elevado, próximo ou superior ao limite recomendado quando considerado o desvio padrão. O mesmo comportamento é observado nos episódios de hipoglicemia nível 1. Esses resultados demonstram a dificuldade do paciente em adaptar-se a rotinas alimentares desorganizadas e ricas em carboidratos, o que compromete o controle glicêmico.

Tabela 22 – Resultados por Cenário - Adolescente 5

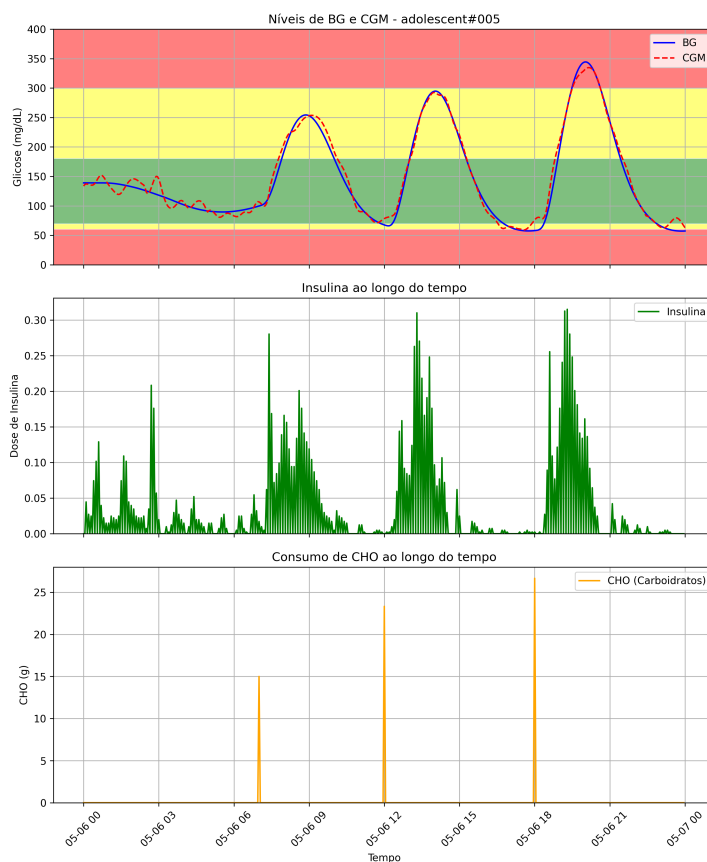
Cenário	MinBLG (mg/dL)	MaxBLG (mg/dL)	Tempo no Alvo (%)	Below L1 (%)	Below L2 (%)	CV (%)
1	57.20	344.28	58.21	13.31	0.00	52.42
2	83.69	340.65	74.84	0.00	0.00	42.65
3	81.77	245.68	77.96	0.00	0.00	27.16
4	87.54	260.74	75.68	0.00	0.00	32.32
5	86.22	250.74	75.26	0.00	0.00	32.49
6	87.33	234.71	77.96	0.00	0.00	29.23
7	96.25	254.49	76.92	0.00	0.00	29.26

Fonte: Autor.

A Figura 30 ilustra esse comportamento glicêmico no Cenário 1, com picos significativos de glicose após as refeições, e pouco tempo de recuperação entre as ingestões alimentares. Em contraste, os demais cenários apresentam melhor distribuição das refeições, favorecendo o controle glicêmico.

Nos testes de estresse, o Adolescente 5 apresentou os resultados mostrados na Tabela 23.

Figura 30 – Gráfico de Glicose do Adolescente 5.



Fonte: Autor.

Tabela 23 – Resultados dos Testes de Estresse do Adolescente 5 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	93.98 ± 7.95	59.94 ± 7.43	55.42 ± 4.17
Maior Glicose (mg/dL)	139.72 ± 1.67	359.86 ± 10.83	463.25 ± 17.08
Hipoglicemia (%)	0	40	80
Hiperglicemia (%)	0	0	100

Fonte: Autor.

O Adolescente 8 conseguiu cumprir todas as metas glicêmicas, conforme a Tabela 24.

Nos testes de estresse, os resultados para o Adolescente 8 estão apresentados na Tabela 25.

6.2.1.3 Adultos

Para a realização dos testes, foram selecionados, arbitrariamente, os adultos 3, 6 e 9. Conforme mostrado na Tabela 26, foram definidos os seguintes valores de TDD e PF

Tabela 24 – Resultados da Primeira Prova de Conceito: Adolescente 8.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	79.10	13.45
Maior Glicose (mg/dL)	274.99	31.60
Porcentagem no Alvo (70–180 mg/dL) (%)	70.39	4.71
Hipo Nível 1 (54–69 mg/dL) (%)	0.86	1.82
Hipo Nível 2 (<54 mg/dL) (%)	0.17	0.58
Coeficiente de Variação (%)	34.86	5.57

Fonte: Autor.

Tabela 25 – Resultados dos Testes de Estresse do Adolescente 8 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	83.27 ± 6.92	48.59 ± 8.09	44.30 ± 3.53
Maior Glicose (mg/dL)	144.30 ± 0.00	358.88 ± 6.09	463.93 ± 17.31
Hipoglicemia (%)	0	100	100
Hiperglicemia (%)	0	0	100

Fonte: Autor.

para cada um deles:

Tabela 26 – Valores de TDD e PF para Diferentes Adultos Simulados.

Adulto	TDD	PF
3	39.0	7
6	23.5	5
9	29	6

Fonte: Autor.

Nos testes padrão, o adulto 3 atingiu os resultados esperados. Seus dados durante o período de teste estão apresentados na Tabela 27.

Tabela 27 – Resultados da Primeira Prova de Conceito: Adulto 3.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	97.77	7.31
Maior Glicose (mg/dL)	199.12	15.43
Porcentagem no Alvo (70–180 mg/dL) (%)	89.16	5.79
Hipo Nível 1 (54–69 mg/dL) (%)	0.00	0.00
Hipo Nível 2 (<54 mg/dL) (%)	0.00	0.00
Coeficiente de Variação (%)	17.89	4.04

Fonte: Autor.

Observa-se que o usuário permaneceu, na maior parte do tempo, dentro da faixa-

alvo. Além disso, ele cumpre as metas estabelecidas relacionadas ao Coeficiente de variação e à Hipoglicemia.

Para os testes de estresse o adulto 3 apresentou os seguintes resultados expostos na Tabela 28.

Tabela 28 – Resultados dos Testes de Estresse do Adulto 3 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	95.14 ± 4.30	81.22 ± 12.08	67.00 ± 10.84
Maior Glicose (mg/dL)	149.17 ± 4.64	215.91 ± 10.25	248.50 ± 16.28
Hipoglicemia (%)	0	0	20
Hiperglicemia (%)	0	0	0

Fonte: Autor.

Pode-se observar que o adulto manteve-se bem estável durante os testes de estresse, apresentando hipoglicemia em apenas 20% dos casos no cenário 7, correspondente à ingestão rápida de CHO.

O adulto 6 atingiu todos os resultados esperados. Seus dados durante o período de teste estão apresentados na Tabela 29.

Tabela 29 – Resultados da Primeira Prova de Conceito: Adulto 6.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	93.79	17.65
Maior Glicose (mg/dL)	235.42	24.16
Porcentagem no Alvo (70–180 mg/dL) (%)	72.29	3.40
Hipo Nível 1 (54–69 mg/dL) (%)	0.00	0.00
Hipo Nível 2 (<54 mg/dL) (%)	0.00	0.00
Coeficiente de Variação (%)	23.61	4.66

Fonte: Autor.

Pode-se observar que esse adulto atingiu os objetivos definidos. No entanto, ao considerar o desvio padrão, existe a possibilidade de que esses resultados estejam fora das metas estabelecidas. Isso ocorre, porque esse paciente apresenta uma maior sensibilidade à insulina em comparação ao Adulto 3. Dessa forma, o controle do fator de personalização requer um refinamento mais preciso para alcançar resultados mais consistentes.

Para os testes de estresse o adulto 6 apresentou os resultados expostos na Tabela 30.

Tabela 30 – Resultados dos Testes de Estresse do Adulto 6 nos Cenários 5, 6 e 7 da Primeira Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	96.16 $\pm$ 12.93	76.21 $\pm$ 11.51	45.67 $\pm$ 8.29
Maior Glicose (mg/dL)	151.63 $\pm$ 12.10	285.11 $\pm$ 13.16	336.72 $\pm$ 15.18
Hipoglicemia (%)	0	10	100
Hiperglicemia (%)	0	0	0

Fonte: Autor.

Percebe-se que o adulto apresentou hiperglicemia no último caso, referente ao consumo de 120 CHO rapidamente. Neste caso, não foi possível evitar a hiperglicemia do paciente.

O adulto 9 também atingiu todos os resultados esperados durante os testes padrão. Seus dados durante o período de teste estão apresentados na Tabela 31.

Tabela 31 – Resultados da Primeira Prova de Conceito: Adulto 9.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	107.61	10.82
Maior Glicose (mg/dL)	226.48	15.92
Porcentagem no Alvo (70–180 mg/dL) (%)	79.80	4.63
Hipo Nível 1 (54–69 mg/dL) (%)	0.00	0.00
Hipo Nível 2 (<54 mg/dL) (%)	0.00	0.00
Coeficiente de Variação (%)	18.36	3.25

Fonte: Autor.

Para os testes de estresse, o adulto 9 apresentou os seguintes resultados descritos na Tabela 32.

Tabela 32 – Resultados dos Testes de Estresse do Adulto 9 nos Cenários 5, 6 e 7

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	68.84 $\pm$ 38.43	84.09 $\pm$ 20.84	46.23 $\pm$ 25.23
Maior Glicose (mg/dL)	204.81 $\pm$ 63.47	190.82 $\pm$ 49.21	250.24 $\pm$ 17.70
Hipoglicemia (%)	40	10	50
Hiperglicemia (%)	0	0	0

Fonte: Autor.

Pode-se observar que este paciente tornou-se hipoglicêmico mais rápido que os outros adultos, o que pode sugerir que seu PF ideal seja menor ou que ele apresente uma sensibilidade maior à insulina.

É importante destacar que, embora os testes realizados com crianças e adolescentes tenham alcançado as metas propostas, os resultados apresentaram pouca margem de

segurança em relação aos limites estabelecidos. Esse comportamento está diretamente relacionado às rotinas alimentares adotadas. Conforme discutido no [Capítulo 2 - Seção 2.1](#), é fundamental que o paciente diabético mantenha uma alimentação adequada e equilibrada ao longo do tratamento, a fim de garantir sua eficácia. No entanto, algumas das rotinas simuladas neste estudo não seguiram tais recomendações. Isso evidencia que, mesmo com a utilização de um controlador automatizado, a adesão a uma rotina alimentar saudável continua sendo essencial para minimizar variações significativas nos níveis de glicose. Tal conclusão é reforçada pelos testes de estresse, especialmente no cenário 7, onde a maioria dos pacientes virtuais não conseguiu evitar episódios de hiperglicemia e hipoglicemia, justamente devido às grandes flutuações glicêmicas causadas por uma alimentação inadequada.

6.2.2 Resultados da Segunda Prova de Conceito

Para a Segunda Prova de Conceito, conforme discutido no [Capítulo 5 - Seção 5.2.2](#), foi conduzido o treinamento de diversas redes neurais, com o objetivo de identificar aquela que mais se aproximasse do desempenho ideal para a resolução do problema proposto. Ao término do processo de treinamento, todas as redes foram submetidas a testes no simulador, sendo selecionada para compor este documento a que apresentou os melhores resultados.

Os experimentos da Segunda Prova de Conceito foram realizados exclusivamente no ambiente do simulador, utilizando-se os mesmos pacientes considerados na Primeira Prova de Conceito, de modo a manter a comparabilidade entre as análises.

6.2.2.1 Crianças

Os resultados obtidos pela criança 1 estão apresentados na Tabela 33:

Tabela 33 – Resultados da Segunda Prova de Conceito: Criança 1.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	68.18	10.59
Maior Glicose (mg/dL)	335.80	40.48
Porcentagem no Alvo (70–180 mg/dL) (%)	73.28	4.04
Hipo Nível 1 (54–69 mg/dL) (%)	2.25	2.74
Hipo Nível 2 (<54 mg/dL) (%)	0.08	0.38
Coeficiente de Variação (%)	39.06	6.41

Fonte: Autor.

Em comparativo com os resultados da Primeira Prova de Conceito, a criança 1 não apresentou melhoras, mantendo os resultados parecidos aos já obtidos.

Já para os testes de estresse, a criança 1 obteve o resultado apresentado na Tabela 34

Tabela 34 – Resultados dos Testes de Estresse da Criança 1 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	88.90 ± 18.57	57.25 ± 5.89	13.06 ± 19.89
Maior Glicose (mg/dL)	159.57 ± 45.69	234.16 ± 37.23	509.02 ± 772.79
Hipoglicemia (%)	1 (10.0%)	7 (70.0%)	9 (90.0%)
Hiperglicemia (%)	0 (0.0%)	1 (10.0%)	1 (10.0%)

Fonte: Autor.

A criança 4 obteve os resultados expostos na Tabela 35.

Tabela 35 – Resultados da Segunda Prova de Conceito: Criança 4.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	70.94	11.23
Maior Glicose (mg/dL)	291.15	39.01
Porcentagem no Alvo (70–180 mg/dL) (%)	72.43	4.12
Hipo Nível 1 (54–69 mg/dL) (%)	1.81	2.50
Hipo Nível 2 (<54 mg/dL) (%)	0.12	0.68
Coefficiente de Variação (%)	33.45	6.92

Fonte: Autor.

Nela, observa-se que esta criança conseguiu cumprir todas as suas metas estipuladas, exceto o coeficiente de variação caso considerado o desvio-padrão.

Já para os testes de estresse, a criança 4 apresentou os resultados expostos na Tabela 36.

Tabela 36 – Resultados dos Testes de Estresse da Criança 4 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	88.90 ± 18.57	57.25 ± 5.89	13.06 ± 19.89
Maior Glicose (mg/dL)	159.57 ± 45.69	234.16 ± 37.23	509.02 ± 772.79
Hipoglicemia (%)	1 (10.0%)	7 (70.0%)	9 (90.0%)
Hiperglicemia (%)	0 (0.0%)	1 (10.0%)	1 (10.0%)

Fonte: Autor.

Nela, a criança conseguiu obter boas métricas, todavia, apresentou hipoglicemia severa na maioria dos casos, isso ocorre devido ao dataset da rede neural treinada não abranger este cenário.

A criança 7 obteve os resultados expostos na Tabela 37.

Tabela 37 – Resultados da Segunda Prova de Conceito: Criança 7.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	128.23	7.10
Maior Glicose (mg/dL)	248.45	15.86
Porcentagem no Alvo (70–180 mg/dL) (%)	63.50	4.15
Hipo Nível 1 (54–69 mg/dL) (%)	0.00	0.00
Hipo Nível 2 (<54 mg/dL) (%)	0.00	0.00
Coeficiente de Variação (%)	19.86	2.10

Fonte: Autor.

Infelizmente, a Inteligência Artificial não conseguiu prever valores ótimos que mantivessem a glicose do paciente dentro da faixa-alvo por mais de 70% do tempo. Esse resultado ocorreu porque o modelo aprendeu a atribuir valores mais conservadores ao fator de personalização (PF), visando controlar a glicose de pacientes mais sensíveis, como os pacientes 1 e 4. Durante a experimentação dos modelos, observaram-se cenários em que o paciente 7 apresentava bom desempenho, porém, nesse mesmo ajuste, os pacientes 1 e 4 eram prejudicados devido a um modelo mais permissivo em relação ao PF.

Já para os testes de estresse, a criança 7 apresentou os resultados expostos na Tabela 38.

Tabela 38 – Resultados dos Testes de Estresse da Criança 7 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	127.91 ± 7.92	107.10 ± 8.31	92.65 ± 7.22
Maior Glicose (mg/dL)	165.31 ± 6.23	322.00 ± 12.06	379.10 ± 12.52
Hipoglicemia (%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Hiperglicemia (%)	0 (0.0%)	10 (100.0%)	10 (100.0%)

Fonte: Autor.

Na Tabela, a criança obteve 100% de hiperglicemia nos cenários 6 e 7, isto ocorre devido a Inteligência Artificial não ter treinado estes cenários, além disso, a rede neural aprendeu a inferir valores menores do que o necessário para este paciente em específico.

6.2.2.2 Adolescentes

Os resultados obtidos pelo adolescente 2 estão apresentados na Tabela 39:

Na tabela, evidencia-se que o adolescente conseguiu cumprir as métricas estabelecidas.

Já para os testes de estresse, o adolescente 2 obteve os resultados apresentados na Tabela 40.

Tabela 39 – Resultados da Segunda Prova de Conceito: Adolescente 2.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	81.45	11.32
Maior Glicose (mg/dL)	228.39	18.14
Porcentagem no Alvo (70–180 mg/dL) (%)	76.78	5.94
Hipo Nível 1 (54–69 mg/dL) (%)	0.47	1.57
Hipo Nível 2 (<54 mg/dL) (%)	0.12	0.80
Coefficiente de Variação (%)	26.86	3.67

Fonte: Autor.

Tabela 40 – Resultados dos Testes de Estresse do adolescente 2 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	77.60 ± 13.63	18.22 ± 5.11	4.68 ± 4.45
Maior Glicose (mg/dL)	152.42 ± 0.01	267.19 ± 6.19	409.07 ± 11.57
Hipoglicemia (%)	1 (10.0%)	10 (100.0%)	10 (100.0%)
Hiperglicemia (%)	0 (0.0%)	0 (0.0%)	10 (100.0%)

Fonte: Autor.

O adolescente 5 obteve resultados distintos dos outros pacientes da mesma faixa etária, como apresentado na Tabela 41:

Tabela 41 – Resultados da Segunda Prova de Conceito: Adolescente 5.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	45.36	23.45
Maior Glicose (mg/dL)	229.58	23.87
Porcentagem no Alvo (70–180 mg/dL) (%)	69.55	11.06
Hipo Nível 1 (54–69 mg/dL) (%)	8.74	7.15
Hipo Nível 2 (<54 mg/dL) (%)	8.83	8.37
Coefficiente de Variação (%)	41.10	9.84

Fonte: Autor.

Infelizmente, esse paciente apresentou uma resposta à insulina mais parecido com o organismo de uma criança, o que ocasionou a alta porcentagem de hipoglicemia no mesmo. Sendo assim, para realizar o equilíbrio, pode-se apenas reduzir o fator escalar do adolescente para 6, permitindo assim com que o mesmo paciente obtenha os resultados apresentados na Tabela 42.

Já para os testes de estresse, foi mantido o fator escalar em 6, garantindo com que o paciente obtenha os resultados apresentados na Tabela 43

Tabela 42 – Resultados da Segunda Prova de Conceito: Adolescente 5.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	74.65	12.96
Maior Glicose (mg/dL)	253.12	28.56
Porcentagem no Alvo (70–180 mg/dL) (%)	78.72	4.51
Hipo Nível 1 (54–69 mg/dL) (%)	1.64	3.70
Hipo Nível 2 (<54 mg/dL) (%)	0.24	1.03
Coefficiente de Variação (%)	33.64	5.36

Fonte: Autor.

Tabela 43 – Resultados dos Testes de Estresse do adolescente 5 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	85.71 ± 8.59	61.45 ± 5.90	43.37 ± 7.93
Maior Glicose (mg/dL)	139.12 ± 0.12	334.30 ± 13.18	428.74 ± 32.16
Hipoglicemia (%)	0 (0.0%)	6 (60.0%)	10 (100.0%)
Hiperglicemia (%)	0 (0.0%)	10 (100.0%)	10 (100.0%)

Fonte: Autor.

Com isso, o adolescente 6 obteve um bom desempenho no cenário 5, todavia, apresentou tanto hipoglicemia quanto hiperglicemia no cenário 6 e 7. Isto ocorre devido ao próprio organismo do paciente que atua com maior sensibilidade à insulina.

Os resultados obtidos pelo adolescente 8 estão apresentados na Tabela 44:

Tabela 44 – Resultados da Segunda Prova de Conceito: Adolescente 8.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	81.99	9.43
Maior Glicose (mg/dL)	260.07	26.06
Porcentagem no Alvo (70–180 mg/dL) (%)	74.86	3.45
Hipo Nível 1 (54–69 mg/dL) (%)	0.52	1.64
Hipo Nível 2 (<54 mg/dL) (%)	0.04	0.43
Coefficiente de Variação (%)	32.66	3.98

Fonte: Autor.

Na tabela, evidencia-se que o adolescente conseguiu cumprir as métricas estabelecidas.

Já para os testes de estresse, o adolescente 8 obteve os resultados apresentados na Tabela 45.

Tabela 45 – Resultados dos Testes de Estresse do adolescente 8 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	80.62 $\pm$ 3.72	51.65 $\pm$ 3.64	45.12 $\pm$ 6.72
Maior Glicose (mg/dL)	144.30 $\pm$ 0.00	317.81 $\pm$ 10.31	458.79 $\pm$ 13.18
Hipoglicemia (%)	0 (0.0%)	10 (100.0%)	10 (100.0%)
Hiperglicemia (%)	0 (0.0%)	10 (100.0%)	10 (100.0%)

Fonte: Autor.

Na Tabela, observa-se que o adolescente não obteve bons resultados nos cenários 6 e 7, isto ocorreu devido a falta de dados inerentes a estes cenários no *dataset* de treinamento.

6.2.2.3 Adultos

Os resultados obtidos pelo adulto 3 estão apresentados na Tabela 46:

Tabela 46 – Resultados da Segunda Prova de Conceito: Adulto 3.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	68.18	10.59
Maior Glicose (mg/dL)	335.80	40.48
Porcentagem no Alvo (70–180 mg/dL) (%)	73.28	4.04
Hipo Nível 1 (54–69 mg/dL) (%)	2.25	2.74
Hipo Nível 2 (<54 mg/dL) (%)	0.08	0.38
Coefficiente de Variação (%)	39.06	6.41

Fonte: Autor.

Na tabela, observa-se que o adulto conseguiu cumprir as métricas estabelecidas.

Já para os testes de estresse, o adulto 3 obteve o resultado exposto na Tabela 47.

Tabela 47 – Resultados dos Testes de Estresse do Adulto 3 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	92.00 $\pm$ 7.56	68.11 $\pm$ 20.05	68.08 $\pm$ 16.32
Maior Glicose (mg/dL)	199.26 $\pm$ 154.50	245.46 $\pm$ 95.53	271.60 $\pm$ 131.91
Hipoglicemia (%)	0 (0.0%)	1 (10.0%)	1 (10.0%)
Hiperglicemia (%)	1 (10.0%)	1 (10.0%)	1 (10.0%)

Fonte: Autor.

Observa-se que o adulto 3 obteve resultados satisfatórios nos testes de estresse, mantendo-se na maioria dos testes dentro da faixa-alvo.

Os resultados obtidos pelo adulto 6 estão apresentados na Tabela 48:

Tabela 48 – Resultados da Segunda Prova de Conceito: Adulto 6.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	71.72	17.88
Maior Glicose (mg/dL)	228.68	22.37
Porcentagem no Alvo (70–180 mg/dL) (%)	78.63	6.35
Hipo Nível 1 (54–69 mg/dL) (%)	1.41	1.82
Hipo Nível 2 (<54 mg/dL) (%)	0.47	1.19
Coefficiente de Variação (%)	27.26	5.30

Fonte: Autor.

Na tabela, observa-se que o adulto 6 conseguiu cumprir as métricas estabelecidas. Todavia, esse adulto apresentou alguns casos de hipoglicemia, isso ocorreu devido à maior fragilidade do mesmo quanto à insulina.

Já para os testes de estresse, o adulto 6 obteve o resultado exposto na Tabela 49.

Tabela 49 – Resultados dos Testes de Estresse do Adulto 6 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	91.49 ± 6.83	75.05 ± 7.06	59.88 ± 13.86
Maior Glicose (mg/dL)	152.07 ± 6.53	222.51 ± 25.32	240.89 ± 35.54
Hipoglicemia (%)	0 (0.0%)	0 (0.0%)	2 (20.0%)
Hiperglicemia (%)	0 (0.0%)	0 (0.0%)	1 (10.0%)

Fonte: Autor.

Observa-se, na Tabela, que o Adulto 6 apresentou resultados negativos apenas no Cenário 7, enquanto nos demais cenários manteve-se dentro da faixa-alvo em três simulações, ainda que com certa flutuação. Esse comportamento sugere que o modelo possui menor capacidade de generalização para condições semelhantes às do Cenário 7. Dessa forma, tal limitação pode ser minimizada ampliando-se o *dataset* com novos casos que representem, de forma mais consistente, as características presentes nesse cenário específico.

Os resultados obtidos pelo adulto 9 estão apresentados na Tabela 50:

Na tabela, observa-se que o adulto conseguiu cumprir as métricas estabelecidas.

Já para os testes de estresse, o adulto 9 obteve o resultado apresentado na Tabela

Tabela 50 – Resultados da Segunda Prova de Conceito: Adulto 9.

Métrica	Média	Desvio Padrão
Menor Glicose (mg/dL)	91.62	10.74
Maior Glicose (mg/dL)	188.72	13.07
Porcentagem no Alvo (70–180 mg/dL) (%)	95.43	3.98
Hipo Nível 1 (54–69 mg/dL) (%)	0.17	0.90
Hipo Nível 2 (<54 mg/dL) (%)	0.02	0.36
Coefficiente de Variação (%)	17.73	3.28

Fonte: Autor.

Tabela 51 – Resultados dos Testes de Estresse do Adulto 9 nos Cenários 5, 6 e 7 da Segunda Prova de Conceito

Métrica	Cenário 5	Cenário 6	Cenário 7
Menor Glicose (mg/dL)	93.35 ± 7.16	54.82 ± 3.80	5.99 ± 4.50
Maior Glicose (mg/dL)	145.14 ± 0.05	222.09 ± 11.30	262.55 ± 8.47
Hipoglicemia (%)	0 (0.0%)	9 (90.0%)	10 (100.0%)
Hiperglicemia (%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Normal (%)	10 (100.0%)	1 (10.0%)	0 (0.0%)

Fonte: Autor.

Infelizmente, o adulto 9 obteve piores valores para os testes de estresse. Isso ocorreu devido ao *dataset* da rede neural ser menos provido de dados que fossem aderentes a esse perfil de teste.

6.3 Avaliação do Plano de Execução

Após a realização dos testes, foi possível observar diferenças significativas quanto à maturidade e à confiabilidade entre as duas Provas de Conceito.

A [Primeira Prova de Conceito](#), embora ainda apresentasse parte do processo de forma mecanizada e heurística, obteve resultados mais consistentes. Tal desempenho pode ser atribuído à maturidade do estudo base, consolidado em três artigos que culminaram em uma solução satisfatória. Entretanto, esta abordagem mostrou-se menos sensível a situações corriqueiras que podem ocorrer com pacientes, o que motivou o desenvolvimento de uma segunda evolução, sendo chamada de Segunda Prova de Conceito..

Por sua vez, a [Segunda Prova de Conceito](#), ainda que tenha apresentado desempenho inferior, conseguiu eliminar grande parte do processo mecanizado, conferindo maior agilidade e usabilidade à execução do *software*. Além disso, essa abordagem apresenta ampla margem de evolução, sobretudo em relação ao *dataset* utilizado e aos próprios modelos de Inteligência Artificial. Ressalta-se ainda o potencial dessa solução para alcançar melhores resultados em cenários de estresse, uma vez que as redes neurais poderiam ser

treinadas para lidar de forma mais eficaz com episódios de hiperglicemia severa, aspecto pouco explorado na [Primeira Prova de Conceito](#). Vale destacar que, com a realização de um maior número de iterações do [Método de Análise de Resultados Proposto](#), é provável que os resultados obtidos por esta Prova de Conceito superassem aqueles verificados na primeira. Entretanto, isso será algo a ser observado em trabalhos futuros..

De modo geral, os resultados obtidos confirmam a viabilidade do controle glicêmico em pacientes com diabetes tipo 1 por meio da integração entre técnicas de aprendizado profundo e lógica difusa. Esses resultados também evidenciam o êxito do plano de ação proposto.

6.4 Resumo do Capítulo

Neste capítulo, apresentou-se o [Plano de Ação](#) adotado para a condução e a análise dos resultados. Na sequência, foi descrita a [Coleta de Dados](#), destacando os resultados obtidos na [Primeira Prova de Conceito](#) e na [Segunda Prova de Conceito](#). Por fim, foi realizada uma [Avaliação do Plano de Execução](#), sintetizando os principais achados e observações decorrentes das simulações. Em resumo, a o pâncreas artificial desenvolvido demonstra viabilidade técnica, bem como acorda resultados interessantes que podem ser ainda mais refinados/aprimorados em abordagens futuras.

7 Conclusão

Este capítulo apresenta o [Contexto Geral](#) que motivou a elaboração deste trabalho; Na sequência, expõe o [Estado Final do Trabalho](#), incluindo o nível de satisfação quanto à [Questão de Pesquisa](#) e do alcance dos [Objetivos](#), apresentando o status do Objetivo Geral e dos Objetivos Específicos apresentando seus respectivos comprobatórios. Por conseguinte, serão evidenciadas as [Atividades Realizadas](#) durante o período completo do trabalho; as [Contribuições](#) que esta pesquisa oferece à sociedade; os possíveis [Futuros Trabalhos](#), os quais promovem a evolução da proposta a partir das limitações identificadas na pesquisa, bem como conferem um norte de como tais limitações podem ser abordadas; e as [Considerações Finais](#) do autor a respeito do trabalho. Por fim, o [Resumo do Capítulo](#) traz um sumário dos principais pontos abordados.

7.1 Contexto Geral

A Diabetes Mellitus é uma doença crônica caracterizada pela deficiência na produção ou na absorção da insulina, hormônio essencial para a regulação da glicose no sangue e para o fornecimento de energia ao organismo ([Ministério da Saúde, 2025](#)). Quando não controlada, essa condição pode levar à hiperglicemia, aumentando significativamente o risco de complicações graves ([EGAN; DINNEEN, 2019](#)) e, em casos extremos, podendo resultar em óbito.

O diabetes tipo 1, foco deste estudo, resulta de um ataque autoimune às células Beta pancreáticas, levando à produção insuficiente ou inexistente de insulina. Sem esse hormônio, a glicose não consegue ser utilizada pelas células como fonte de energia, acumulando-se no sangue e gerando consequências metabólicas graves. Como resultado, torna-se essencial a reposição exógena do hormônio de maneira vitalícia ([ATKINSON; EISENBARTH; MICHELS, 2014](#)).

Ao longo dos anos, os métodos de gerenciamento do diabetes tipo 1 têm evoluído significativamente ([DIMEGLIO; EVANS-MOLINA; ORAM, 2018](#)), chegando ao patamar onde é possível realizar o controle da glicemia através de um dispositivo conhecido como Pâncreas Artificial, no qual é a combinação entre uma bomba de insulina, um CGM e um controlador, tudo isso em um Sistema de Malha Fechada ([COBELLI; RENARD; KOVATCHEV, 2011a](#)).

Com isso, torna-se possível construir um controlador baseado em lógica difusa e integrá-lo a uma rede neural, com a finalidade de promover um controle mais eficaz da glicemia do paciente, de modo que o dispositivo seja sensível a mudanças repentinas em

sua rotina.

7.2 Estado Final do Trabalho

Esta seção está dividida em duas subseções, sendo a primeira para a apresentação das respostas ao questionamento especificado no [Capítulo 1](#) dessa monografia; e a segunda para explorar o cumprimento dos objetivos, seja Objetivo Geral, sejam Objetivos Específicos.

7.2.1 Questão de Pesquisa

Este trabalho apresenta como [Questão de Pesquisa](#) o seguinte questionamento:

É possível desenvolver, *in silico*, um sistema de controle de glicose que proporcione maior comodidade ao usuário, minimizando a necessidade de inserção manual de informações pessoais?

Como evidenciado pelo [Capítulo 6](#), é possível desenvolver um sistema de controle de glicose com as características requisitadas pela questão de pesquisa. No entanto, faz-se necessário o bom senso do paciente quanto à rotina para o bom funcionamento do produto.

Sendo assim, na hipótese central, de que "um pâncreas artificial proporciona maior comodidade ao usuário, minimizando a necessidade de inserção manual de informações pessoais", percebe-se que a proposta confere viabilidade técnica, já conseguindo resultados satisfatórios em boa parte dos casos testados. Todavia, ainda são necessários estudos mais aprofundados, bem como evoluções na proposta, em especial no *dataset*, visando maior estabilidade nos resultados, abrangência de casos com adequados resultados, para enfim proporcionar maior comodidade ao usuário.

7.2.2 Objetivos

O cumprimento dos [Objetivos](#) desta pesquisa é evidenciado a seguir, sendo organizados como Geral e Específicos.

7.2.2.1 Objetivo Geral

O [Objetivo Geral](#) deste trabalho consiste no desenvolvimento, *in silico*, de um controlador capaz de gerenciar a glicemia de forma personalizada, adaptando-se às necessidades específicas de cada paciente.

Em consonância com a [Questão de Pesquisa](#), verifica-se que o objetivo geral foi alcançado, sendo possível desenvolver um pâncreas artificial com as condições especificadas.

7.2.2.2 Objetivos Específicos

Os **Objetivos Específicos** estão apresentados a seguir, sendo todos eles cumpridos. Segue a lista dos mesmos, bem como seus status e comprobatórios:

1. Estudo sobre lógica difusa;
2. Estudo sobre Aprendizado de Máquina e Aprendizado Profundo;
3. Estudo sobre sistemas manuteníveis, os quais detenham a capacidade de atender a mudanças contínuas ao longo do seu ciclo de vida;
4. Desenvolvimento de provas de conceito, considerando os conhecimentos adquiridos nos estudos anteriores, visando uma solução minimalista, embasada e testada; e
5. Documentação dos resultados obtidos, de forma a evidenciar a personalização e a adaptação da solução diante de diferentes necessidades dos pacientes, o que confere maior comodidade aos mesmos.

A Tabela 52 apresenta o número do objetivo, seu status de cumprimento e o respectivo comprobatório.

Tabela 52 – Status e Comprobatório dos Objetivos Específicos.

Objetivo	Status	Comprobatório
1	Cumprido	Documentado na Seção 2.2
2	Cumprido	Documentado na Seção 2.3
3	Cumprido	Apresentado nas Seções 2.4 e 2.5
4	Cumprido	Detalhado no Capítulo 5
5	Cumprido	Documentado no Capítulo 6

Fonte: Autor.

7.3 Atividades Realizadas

As atividades desenvolvidas na primeira etapa do TCC foram apresentadas na seção [Atividades da Primeira Etapa do TCC](#), na qual são descritas com seus respectivos *status*, períodos e resultados. Já as atividades desenvolvidas na segunda etapa do TCC foram apresentadas na seção [Atividades da Segunda Etapa do TCC](#), na qual são descritas com seus respectivos *status*, períodos e resultados.

7.4 Contribuições

Este trabalho oferece algumas contribuições para a comunidade acadêmica e profissional os quais merecem menção. Dentre elas, cabe destacar que esta pesquisa:

1. Demonstra a implementação de um pâncreas artificial por meio de um simulador, contribuindo para um ambiente de desenvolvimento mais seguro e controlado. Essa abordagem possibilita a realização de testes sem riscos a pacientes reais, promovendo maior confiabilidade nos resultados obtidos;
2. Apresenta, de forma detalhada, todas as etapas envolvidas no desenvolvimento do pâncreas artificial¹, oferecendo uma documentação que pode servir como referência técnica e metodológica para futuros estudos em Engenharia Biomédica;
3. Analisa critérios de qualidade para o software com base em métricas pre-estabelecidas por autoridades médicas competentes;
4. Fornece uma explicação aprofundada sobre os conceitos de lógica difusa e redes neurais artificiais, consolidando fundamentos teóricos que sustentam o desenvolvimento do sistema proposto e que podem orientar futuras pesquisas na área; e
5. Propõe reflexões relevantes sobre a Diabetes Mellitus, abordando aspectos clínicos, tecnológicos e sociais, contribuindo para a compreensão mais ampla da doença e seus impactos, especialmente no contexto da aplicação de tecnologias assistivas.

Estas contribuições podem servir como base para futuros trabalhos na área de engenharia biomédica, bem como auxiliar no desenvolvimento de *software* para fins médicos.

7.5 Futuros Trabalhos

Este trabalho apresenta, também, algumas limitações que podem ser exploradas em trabalhos futuros. Dentre elas, destacam-se:

1. Ampliação da base de dados por meio da obtenção de informações adicionais relacionadas aos fatores de personalização ideais, com o objetivo de aprimorar o desempenho e a generalização da rede neural proposta;
2. Desenvolvimento de um novo modelo de Inteligência Artificial, voltado à predição direta da dose de insulina, buscando integrar em um único sistema a personalização e a recomendação terapêutica de forma mais eficaz;
3. Implementação de um sistema embarcado dedicado à execução do software do controlador já desenvolvido, visando maior portabilidade, autonomia e viabilidade para uso em tempo real no contexto clínico;

¹ Repositório disponível em: <https://github.com/kaue-pt/TCC_ArtificialPancreas>. Acesso em: 06 nov. 2025.

4. Criação de um aplicativo para dispositivos móveis, com a finalidade de permitir o monitoramento contínuo e em tempo real dos níveis de glicemia do paciente usuário do pâncreas artificial, promovendo maior acessibilidade e segurança na gestão da doença; e
5. Adoção de uma arquitetura baseada em Sistemas Multiagentes, permitindo maior personalização dos parâmetros individuais e favorecendo o processamento paralelo. Nessa abordagem, as inferências da rede neural são delegadas a um servidor central, otimizando o desempenho do sistema e reduzindo a carga computacional do dispositivo embarcado utilizado pelo paciente.

Estas sugestões podem servir como ponto de partida para pesquisas futuras no desenvolvimento de um Pâncreas Artificial.

7.6 Considerações Finais

O autor considera que os resultados alcançados durante o Trabalho de Conclusão de Curso foram satisfatórios, uma vez que o Objetivo Geral foi atendido e os Objetivos Específicos foram considerados como cumpridos. Ademais, toda a documentação planejada foi desenvolvida com êxito. Todavia, o trabalho ainda não pode ser considerado concluído, pois, como evidenciado na seção de [Trabalhos Futuros](#), há ainda diversas possibilidades de aprimoramento do produto aqui descrito.

7.7 Resumo do Capítulo

Neste capítulo, foi discutido a respeito do [Contexto Geral](#) do trabalho no qual fundamentou a pesquisa. Na sequência, discutiu-se a respeito do [Estado Final do Trabalho](#), evidenciando a [Questão de Pesquisa](#) e o *status* dos [Objetivos](#). O capítulo também ressalta sobre as [Atividades Realizadas](#) ao longo do TCC; as [Contribuições](#) de maior relevância fornecidas para a comunidade científica e profissional; os [Futuros Trabalhos](#) identificados; e por fim as [Considerações Finais](#) do autor a respeito deste trabalho.

Referências

- AMAZON. *O que é aprendizado profundo? – Explicação sobre IA de aprendizado profundo – AWS*. 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/deep-learning/>>. Citado na página 52.
- ATKINSON, M. A.; EISENBARTH, G. S.; MICHELS, A. W. Type 1 diabetes. *The lancet*, Elsevier, v. 383, n. 9911, p. 69–82, 2014. Citado 2 vezes nas páginas 30 e 135.
- BANSAL, H. O.; SHARMA, R.; SHREERAMAN, P. Pid controller tuning techniques: a review. *Journal of control engineering and technology*, Arastirmax Scientific Publication Index, v. 2, n. 4, 2012. Citado na página 39.
- BOCHIE, K. et al. Aprendizado profundo em redes desafiadoras: Conceitos e aplicações. *XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, cap. 4, 2020. Citado 7 vezes nas páginas 52, 53, 54, 56, 57, 58 e 59.
- BOEG, J. Kanban em 10 passos. *Tradução de Leonardo Campos, Marcelo Costa, Lúcio Camilo, Rafael Buzon, Paulo Rebelo, Eric Fer, Ivo La Puma, Leonardo Galvão, Thiago Vespa, Manoel Pimentel e Daniel Wildt. C4Media*, p. 27, 2010. Citado 2 vezes nas páginas 80 e 81.
- CHEN, G.; PHAM, T. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press, 2000. ISBN 9781420039818. Disponível em: <<https://books.google.com.br/books?id=ToZyn99N6GwC>>. Citado 3 vezes nas páginas 43, 45 e 79.
- COBELLI, C.; RENARD, E.; KOVATCHEV, B. Artificial pancreas: past, present, future. *Diabetes*, American Diabetes Association, v. 60, n. 11, p. 2672–2682, 2011. Citado 2 vezes nas páginas 31 e 135.
- COBELLI, C.; RENARD, E.; KOVATCHEV, B. Artificial pancreas: Past, present, future. *Diabetes*, v. 60, n. 11, p. 2672–2682, 10 2011. ISSN 0012-1797. Disponível em: <<https://doi.org/10.2337/db11-0654>>. Citado na página 33.
- COCKBURN, A. *Alistair.Cockburn.us*. 2009. Disponível em: <<https://web.archive.org/web/20140329201018/http://alistair.cockburn.us/Hexagonal+architecture>>. Citado 2 vezes nas páginas 66 e 67.
- COLAB, G. *Google Colab*. 2025. Disponível em: <<https://colab.research.google.com/>>. Citado na página 71.
- DIMEGLIO, L. A.; EVANS-MOLINA, C.; ORAM, R. A. Type 1 diabetes. *The Lancet*, Elsevier, v. 391, n. 10138, p. 2449–2462, 2018. Citado 3 vezes nas páginas 30, 31 e 135.
- DOCS, G. *Documentos Google*. 2025. Disponível em: <<https://docs.google.com/document/u/0/?hl=pt-BR>>. Citado na página 69.
- DRUMOND, R. R. et al. An lstm recurrent network for motion classification from sparse data. In: *Proceedings of the 13th international joint conference on computer vision*,

imaging and computer graphics theory and applications. [S.l.: s.n.], 2018. v. 1, p. 215–22. Citado na página 60.

EGAN, A. M.; DINNEEN, S. F. What is diabetes? *Medicine*, v. 47, n. 1, p. 1–4, 2019. ISSN 1357-3039. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1357303918302627>>. Citado 4 vezes nas páginas 29, 31, 33 e 135.

ENGEL, G. I. Pesquisa-ação. *Educar em revista*, SciELO Brasil, p. 181–191, 2000. Citado na página 82.

EVANS, E. *Domain-driven design: tackling complexity in the heart of software*. [S.l.]: Addison-Wesley Professional, 2004. Citado 2 vezes nas páginas 65 e 66.

FONSECA, J. J. S. da. *Apostila de metodologia da pesquisa científica*. [S.l.]: João José Saraiva da Fonseca, 2002. Citado na página 75.

FLOWER, M. *Inversion of Control Containers and the Dependency Injection pattern*. 2004. Disponível em: <<https://www.martinfowler.com/articles/injection.html>>. Citado na página 67.

FLOWER, M. *Mocks Aren't Stubs*. 2007. Disponível em: <<https://martinfowler.com/articles/mocksArentStubs.html>>. Citado 2 vezes nas páginas 65 e 67.

FLOWER, M. *bliki: Domain Driven Design*. 2020. Disponível em: <<https://martinfowler.com/bliki/DomainDrivenDesign.html>>. Citado 2 vezes nas páginas 65 e 66.

FRONTONI, S. et al. Glucose variability: An emerging target for the treatment of diabetes mellitus. *Diabetes Research and Clinical Practice*, v. 102, n. 2, p. 86–95, 2013. ISSN 0168-8227. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168822713003227>>. Citado na página 101.

GAMMA, E. et al. *Design patterns: elements of reusable object-oriented software*. USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN 0201633612. Citado 2 vezes nas páginas 67 e 102.

GEEKSFORGEEEKS. Sliding window technique. *GeeksforGeeks*, abr. 2017. Disponível em: <<https://www.geeksforgeeks.org/window-sliding-technique/>>. Citado na página 95.

GERHARDT, T.; SILVEIRA, D. *Metodos de Pesquisa*. Plageder, 2009. (Serie Educacao a Distancia - UFRGS). ISBN 9788538600718. Disponível em: <<https://books.google.com.br/books?id=dRuzRyEIzmkC>>. Citado 2 vezes nas páginas 75 e 76.

GIL, A. *Como elaborar projetos de pesquisa*. Atlas, 2002. ISBN 9788522431694. Disponível em: <<https://books.google.com.br/books?id=X4uvAAAACAAJ>>. Citado na página 76.

GIT. *Git*. 2025. Disponível em: <<https://git-scm.com/>>. Citado na página 71.

GITHUB. *Build software better, together*. 2025. Disponível em: <<https://github.com/>>. Citado na página 71.

- GÓES, A. P. P.; VIEIRA, M. R. R.; JÚNIOR, R. D. R. L. Diabetes mellitus tipo 1 no contexto familiar e social. *Revista paulista de pediatria*, SciELO Brasil, v. 25, p. 124–128, 2007. Citado na página 31.
- GOMES, D. T. Redes neurais recorrentes para previsão de séries temporais de memórias curta e longa. *Master's thesis, Department of Statistics, Campinas State University, Campinas, Brazil*, p. 153, 2005. Citado na página 59.
- GOMIDE, F.; GUDWIN, R. R.; TANSCHKEIT, R. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. In: *Proc. 6 th IFSA Congress-Tutorials*. [S.l.: s.n.], 1995. v. 59, p. 1–38. Citado 2 vezes nas páginas 47 e 49.
- GOMIDE, F. A. C.; GUDWIN, R. R. Modelagem, controle, sistemas e lógica fuzzy. *SBA controle & Automação*, v. 4, n. 3, p. 97–115, 1994. Citado 2 vezes nas páginas 49 e 50.
- GRAVES, A. et al. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 5, p. 855–868, 2009. Citado na página 60.
- GÜRSEL, G. Healthcare, uncertainty, and fuzzy logic. *Digital Medicine*, LWV, v. 2, n. 3, p. 101–112, 2016. Citado 2 vezes nas páginas 31 e 32.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. Bookman Editora, 2001. ISBN 9788577800865. Disponível em: <<https://books.google.com.br/books?id=bhMwDwAAQBAJ>>. Citado 6 vezes nas páginas 32, 53, 54, 56, 57 e 79.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural Computation*, v. 9, p. 1735–1780, 11 1997. Citado na página 59.
- HUNT, K. et al. Neural networks for control systems—a survey. *Automatica*, v. 28, n. 6, p. 1083–1112, 1992. ISSN 0005-1098. Disponível em: <<https://www.sciencedirect.com/science/article/pii/000510989290053I>>. Citado na página 41.
- IANCU, I. A mamdani type fuzzy logic controller. *Fuzzy logic-controls, concepts, theories and applications*, InTech London, UK, v. 15, n. 2, p. 325–350, 2012. Citado 2 vezes nas páginas 51 e 52.
- KLIR, G.; YUAN, B. *Fuzzy sets and fuzzy logic*. [S.l.]: Prentice hall New Jersey, 1995. v. 4. Citado 3 vezes nas páginas 42, 43 e 44.
- KOPIER, A. A. et al. Redes neurais artificiais e suas aplicações no setor elétrico. *Revista de Engenharias da Faculdade Salesiana*, v. 1, n. 9, p. 27–33, 2019. Citado na página 54.
- KOTTAPALLE, P. et al. Poc design: A methodology for proof-of-concept (poc) development on internet of things connected dynamic environments. *Security and Communication Networks*, v. 2021, p. 1–12, 10 2021. Citado na página 76.
- KOTTAPALLE, P. et al. Poc design: A methodology for proof-of-concept (poc) development on internet of things connected dynamic environments. *Security and Communication Networks*, v. 2021, p. 1–12, 10 2021. Citado 2 vezes nas páginas 79 e 82.
- LATEX. *LaTeX*. 2025. Disponível em: <<https://www.latex-project.org/>>. Citado na página 72.

- LAZAR, S. et al. How to measure glycemic variability? a literature review. *Medicina*, v. 60, n. 1, 2024. ISSN 1648-9144. Disponível em: <<https://www.mdpi.com/1648-9144/60/1/61>>. Citado na página 42.
- LUCIDCHART. *Blog*. 2025. Disponível em: <<https://www.lucidchart.com/blog>>. Citado na página 72.
- MAN, C. D. et al. The uva/padova type 1 diabetes simulator: New features. *Journal of Diabetes Science and Technology*, Diabetes Technology Society, v. 8, n. 1, p. 26–34, 2014. ISSN 1932-2968. Submitted to FDA in 2013 as S2013. Improvements include nonlinear insulin action during hypoglycemia, glucagon kinetics, and refined virtual patient generation. Disponível em: <<https://doi.org/10.1177/1932296813514502>>. Citado na página 70.
- MARTINS, M. E. G. Coeficiente de determinação. *Revista Ciência Elementar*, v. 6, n. 1, p. 24, 2018. Citado na página 64.
- MATLAB. *What Is Model Predictive Control?* 2025. Disponível em: <<https://www.mathworks.com/help/mpc/gs/what-is-mpc.html>>. Citado na página 40.
- MATSUNAGA, V. Y. Curso de redes neurais utilizando o matlab. *Belém do Pará*, 2012. Citado na página 57.
- MAUSETH, R. et al. Use of a “fuzzy logic” controller in a closed-loop artificial pancreas. *Diabetes technology & therapeutics*, Mary Ann Liebert, Inc. 140 Huguenot Street, 3rd Floor New Rochelle, NY 10801 USA, v. 15, n. 8, p. 628–633, 2013. Citado 8 vezes nas páginas 41, 51, 52, 79, 94, 98, 99 e 115.
- MAUSETH, R. et al. Stress testing of an artificial pancreas system with pizza and exercise leads to improvements in the system’s fuzzy logic controller. *Journal of diabetes science and technology*, SAGE Publications Sage CA: Los Angeles, CA, v. 9, n. 6, p. 1253–1259, 2015. Citado 4 vezes nas páginas 79, 94, 98 e 115.
- MAUSETH, R. et al. Proposed clinical application for tuning fuzzy logic controller of artificial pancreas utilizing a personalization factor. *Journal of diabetes science and technology*, SAGE Publications, v. 4, n. 4, p. 913–922, 2010. Citado 10 vezes nas páginas 31, 41, 79, 91, 94, 97, 100, 101, 114 e 115.
- MICROSOFT. *Visual Studio Code*. 2021. Disponível em: <<https://code.visualstudio.com/docs>>. Citado na página 71.
- MING, W. et al. Recent advances in the precision control strategy of artificial pancreas. *Medical & Biological Engineering & Computing*, Springer, v. 62, n. 6, p. 1615–1638, 2024. Citado 5 vezes nas páginas 39, 40, 41, 79 e 91.
- Ministério da Saúde. *Diabetes*. 2025. Acesso em: 7 abr. 2025. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/diabetes>>. Citado 2 vezes nas páginas 29 e 135.
- MKPB. *Diferenças entre estudos in vitro, in vivo e in silico (MPKB)*. 2015. Disponível em: <https://mpkb.org/home/patients/assessing_literature/in_vitro_studies>. Citado na página 32.

NAGAMINE, R. K. Lógica difusa para controle não convencional de uma viga inteligente. *São Carlos*, 2001. Citado na página 42.

NISE, N. *Engenharia De Sistemas De Controle*. LTC, 2023. ISBN 9788521621355. Disponível em: <<https://books.google.com.br/books?id=ZsXINAEACAAJ>>. Citado 3 vezes nas páginas 35, 37 e 38.

NOSOUHIAN, S.; NOSOUHIAN, F.; KHOSHOUEI, A. K. A review of recurrent neural network architecture for sequence learning: Comparison between lstm and gru. Preprints, 2021. Citado na página 62.

OBSIDIAN. *Meet the teams that trust Obsidian*. 2025. Disponível em: <<https://obsidian.md/enterprise/>>. Citado na página 69.

OGATA, K. *Engenharia de Controle Moderno*. Pearson Universidades, 2010. ISBN 9788576058106. Disponível em: <<https://books.google.com.br/books?id=iL3FYgEACAAJ>>. Citado 3 vezes nas páginas 35, 37 e 38.

OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Citado 3 vezes nas páginas 60, 61 e 62.

OVERLEAF. *Overleaf*. 2025. Disponível em: <<https://www.overleaf.com>>. Citado na página 72.

PITITTO, B. et al. *Metas no tratamento do diabetes*. [S.l.]: Sociedade Brasileira de Diabetes, 2023. Diretriz Oficial da Sociedade Brasileira de Diabetes. ISBN 978-85-5722-906-8. Citado na página 42.

PYTHON. *3.13.2 Documentation*. 2025. Disponível em: <<https://docs.python.org/3/>>. Citado na página 70.

READER, A. A. *Fazer download do Adobe Acrobat Reader: visualizador gratuito de PDFs*. 2025. Disponível em: <<https://get.adobe.com/br/reader/>>. Citado na página 69.

SAÚDE, M. da. 5 dicas para cuidar da alimentação de quem possui diabetes. *Ministério da Saúde*, fev. 2022. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/saude-brasil/eu-quero-me-alimentar-melhor/noticias/2022/5-dicas-para-cuidar-da-alimentacao-de-quem-possui-diabetes>>. Citado na página 42.

SHEETS, G. *Planilhas Google*. 2025. Disponível em: <<https://docs.google.com/spreadsheets/u/0/>>. Citado na página 71.

SHIRI, F. M. et al. A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru. *arXiv preprint arXiv:2305.17473*, 2023. Citado 2 vezes nas páginas 62 e 63.

SIGURÐARDÓTTIR, Á. K. Self-care in diabetes: model of factors affecting self-care. *Journal of clinical nursing*, Wiley Online Library, v. 14, n. 3, p. 301–314, 2005. Citado na página 32.

SILVA, F. F. B. da. *Desvendando a Lógica Difusa*. Dissertação (Dissertação (Mestrado)) — Universidade Federal de Uberlândia, Uberlândia, Brasil, 2011. Citado na página 43.

- SINGH, D. *Mastering Tanh: A Deep Dive into Balanced Activation for Machine Learning*. 2024. Disponível em: <<https://medium.com/ai-enthusiast/mastering-tanh-a-deep-dive-into-balanced-activation-for-machine-learning-4734ec147dd9>>. Citado na página 56.
- SLACK. *Aqui o trabalho acontece*. 2025. Disponível em: <<https://slack.com/intl/pt-br/>>. Citado na página 72.
- SPÖRL, C.; CASTRO, E.; LUCHIARI, A. Aplicação de redes neurais artificiais na construção de modelos de fragilidade ambiental. *Geography Department, University of Sao Paulo*, p. 113–135, 01 2011. Citado na página 53.
- SÍRIO-LIBANÊS, H. *Hospital Sírio-Libanês*. 2025. Disponível em: <<https://hospitalsiriolibanes.org.br/blog/endocrinologia/a-importancia-de-saber-como-agem-e-quanto-tempo-duram-as-insulinas>>. Citado na página 99.
- TANSCHKEIT, R. Sistemas fuzzy. *Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro*, p. 338–353, 2004. Citado 7 vezes nas páginas 44, 46, 47, 48, 49, 50 e 51.
- TEAMS. *Acesse sua conta*. 2025. Disponível em: <<https://www.microsoft.com/pt-br/microsoft-teams/log-in>>. Citado na página 72.
- TELEGRAM. *Telegram*. 2025. Disponível em: <<https://web.telegram.org/a/>>. Citado na página 72.
- TRELLO. *TCC*. 2025. Disponível em: <<https://trello.com/b/D22DvkOf/tcc>>. Citado na página 69.
- TSIOULI, E. et al. Effects of diabetes-related family stress on glycemic control in young patients with type 1 diabetes: Systematic review. *Canadian Family Physician*, The College of Family Physicians of Canada, v. 59, n. 2, p. 143–149, 2013. Citado na página 32.
- VARGAS, A. C. G.; PAES, A.; VASCONCELOS, C. N. Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In: SN. *Proceedings of the xxix conference on graphics, patterns and images*. [S.l.], 2016. v. 1, n. 4. Citado na página 58.
- WANG, L. *Model Predictive Control System Design and Implementation Using MATLAB®*. Springer London, 2009. (Advances in Industrial Control). ISBN 9781848823303. Disponível em: <<https://books.google.com.br/books?id=PphumLcKPi4C>>. Citado na página 40.
- WANG, W.; LU, Y. Analysis of the mean absolute error (mae) and the root mean square error (rmse) in assessing rounding model. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 324, n. 1, p. 012049, mar 2018. Disponível em: <<https://dx.doi.org/10.1088/1757-899X/324/1/012049>>. Citado na página 64.
- WHO, W. H. O. Diabetes. *World Health Organization: WHO*, nov. 2024. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/diabetes>>. Citado na página 29.

- XIE, S. v. J. *simglucose/README.md at master · jxx123/simglucose*. 2018. Disponível em: <<https://github.com/jxx123/simglucose/blob/master/README.md>>. Citado na página 70.
- YU, Y. et al. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 31, n. 7, p. 1235–1270, 2019. Citado na página 60.
- ZADEH, L. A. Fuzzy sets. *Information and control*, Elsevier, v. 8, n. 3, p. 338–353, 1965. Citado 5 vezes nas páginas 31, 42, 43, 45 e 79.